THE BEST SELLING COMPUTER PROJECTS MAGAZINE JULY 1985

ELECTRONICS & COMPUTING

Books 6

8 Print Firm Collapse

ARA 9 Strategy Trading

15

20

27 Net Numbers

28

38

60 PCB

64 Repair

AN EMAP PUBLICATION

USA $2.95
Germany D6.00
Singapore S$4.95
95p

INCLUDING
QL SOFTWARE DIRECTORY

·SCOOP·

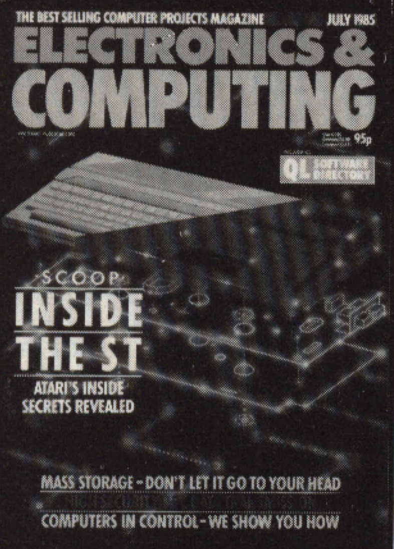
INSIDE THE ST

ATARI'S INSIDE SECRETS REVEALED

MASS STORAGE - DON'T LET IT GO TO YOUR HEAD

HIGH SECURITY BBC DATA TRANSMISSION

COMPUTERS IN CONTROL - WE SHOW YOU HOW



COVER PHOTO BY ROB BRIMSON

Electronics & Computing Monthly
Priory Court, 30-32 Farringdon Lane,
London, EC1R 3AU

Editorial 01-251-6222
Editor Gary Evans
Deputy Editor William Owen

Advertising 01-251-6222
Advertisement Manager Tony Herman
Advertising Executive Tracy Keighley
Advertising Production Serena Hadley

Production 01-251-6222
Art Editor Jeremy Webb
Make-up Time Graphics
Publisher Terry Pratt

Distribution
EMAP National Publications

Published by
EMAP Business and
Computer Publications

Printed by
Riverside Press, England

Subscriptions and Back Issues
Subscriptions
please telephone 01-251 6222
for details.
Back Issues telephone 0858 34567

Electronics & Computing Monthly is normally published on the 13th day of each month.

ABC

ELECTRONICS & COMPUTING Contents

Vol. 5 Issue 7

SOFTWARE

PROJECTS

FEATURES

REVIEWS

PLUS

There are going to be some changes at Electronics and Computing. For more details turn to page 26.

An excellent machine but is it ready for the user?

As the exclusive look at the hardware of the Atari 520ST elsewhere in this issue proves, the ST computer does exist in a final production form. Our information is that approved circuit drawings, PCB layouts and engineering drawings of the casing are at Atari's production facility in the Far East. Given these facts there seems little doubt that, barring accidents, the ST machines will be here in the near future.

Our analysis of the computer's circuitry shows that for the most part the design of the machine is extremely elegant. It points the way to computers of the future – extensive use of ULAs has led to a design that features fewer ICs than many eight-bit micros, yet offers a performance that far outstrips them.

The only possible sign of a rush to get the hardware into production is the fact that, according to our sources, the 520ST will not have an RF modulator allowing an ordinary TV set to be used as the computer's monitor. Instead users will have to purchase one of a range of dedicated monitors from the company.

But this policy could equally have been taken on grounds of the flexibility it offers in terms of production. Amstrad has had occasion to appreciate the fact that the designers of their CPC computers took a similar approach. Machines lying around in UK warehouses during recent months could be shipped to France to meet an upturn in demand there. The different TV standards that prevent this sort of reallocation of stock did not apply in Amstrad's case: PAL, SECAM – it made no difference. Atari too could benefit from this flexibility. There will be no need to modify machines according to the country they are destined for. The only differences that will have to be catered for are variations in the mains voltages in various countries.

If the hardware side of the ST series is sown up, there are still many question marks over the software. Those reviews of the computer that have so far appeared in the UK Press were all rather vague when it came to GEM and the Basic and Logo that are to be bundled with the 520.

Atari has openly admitted that the decision to supply the software on disk was taken in order that any bugs discovered by early users of the machines could be easily fixed. In this they have learnt from Sinclair., The spectacle of endless dongles will not apply to the ST computers.

The fact that the software is unlikely to be fully implemented on early machines gives a clue as to how the machine will initially be marketed. Atari is saying that it certainly sees the computer selling to small businesses as well as to the enthusiast computer user. But the fact that the software will not be finalised on early machines would though seem to rule out any sales in the business area. The ST will be an enthusiasts machine for the first few months. These early users of the machine will be relied upon to discover any shortcomings.

GARY EVANS

Distributor refuses Atari deal

Following the collapse of Prism, Terry Blood Distribution is now the sole distributor of Sinclair hardware and software products.

The company recently issued a statement saying that while TBD has no connection with Prism and has no legal responsibility to resolve any dispute between Prism and its customers, they are in fact assisting those customers who previously dealt with Prism.

The company is fully committed to supporting Sinclair Research and has expanded its trade enquiry section. TBD report an upturn in demand for the QL but also states that returns of Spectrum Plus computers are running at around 7,000 a week!

TBD also recently announced that it was ceasing distribution of the Atari range. The company had a distribution 'arrangement' with Atari for some eight months but had been pressing for a formalisation of the relationship. When TBD received the documentation from Atari that outlined the terms of business they quickly branded them as 'unworkable' and 'commercial suicide'. A spokesman said 'The conditions demanded by Atari in terms of margins and stocking commitment were quite simply unworkable . . . Atari has a good product and we took the decision to cease distribution with some reluctance'.

With the ST computer due in this country in the near future Atari will need as much good will as possible from the large distribution and retail chains if the machine is to get off to a good start. It remains to be seen whether or not other organisations will find Atari's conditions of trading as unworkable as Terry Blood Distribution.

Memory expansion for BBC minus and plus

JP Designs has launched two memory expansion boards for the BBC micro. The first is a sideways RAM/emulator that provides 16K of static RAM. The additional memory is located between 8000 and BFFF in the computer's system memory map. The RAM can be written to directly: a facility that allows machine code software to be placed in the RAM area.

This feature enables the BBC micro plus RAM unit to be used to develop software for small stand alone control systems. A switch on top of the unit can be used to determine whether the BBC micro or the system under development has access to the additional RAM.

The ROM extension board from JP Designs is one of the cheapest on the market. It allows up to eight 28 pin ROMs to be used in the sideways area of the BBC computer.

The ROM expansion board costs £24.95 plus VAT, the RAM/emulator £99.95 plus VAT. *JP Designs, 37 Oyster Row, Cambridge CB5 8LJ.*

Silicon Express gets inside story on QL

Silicon Express, best known as distributors for Microvitec, Philips and Cumana, has launched a disk interface for the Sinclair QL.

The double density interface is the first in a range of QL addons to be given the generic name of 'Insider Boards'. The name reflects the fact that the disk interface is small enough to fit inside the case of the QL.

The interface can be used in conjunction with 5", 3.5" and 5.25" drives. The firmware of the system was written by the author of QDOS and complete compatibility with the QL's operating system should therefore be assured.

The disk interface costs £99 plus VAT. Silicon Express also sells bundled systems. The interface plus single 80 track double sided drive is available at £249, a double drive costs £399 and offers up to 1.4 megabytes of formatted storage.

Silicon Express are at Silicon House, Fowke Street, Rothley, Leicestershire, LE7 7PJ. Telephone (0533) 374917.

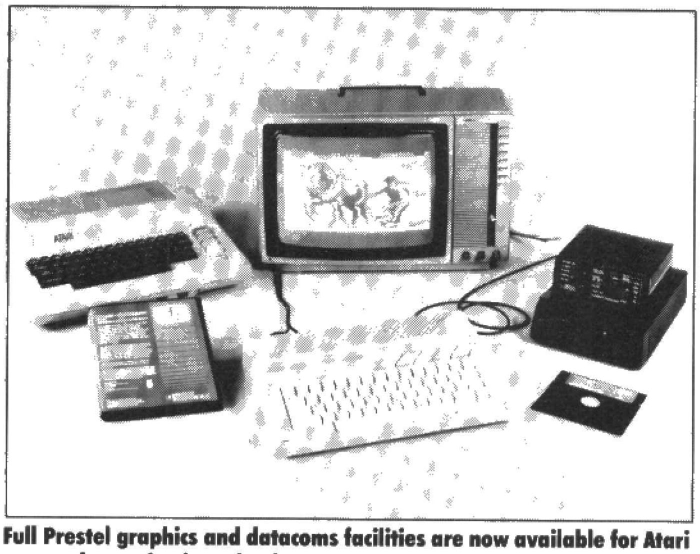
Le Quantum leurp

A French version of the QL computer was launched at the Paris Sicob show in mid May. Faced with a flat UK market, Sinclair Research is looking to Europe in an effort to lift sales of the QL. 8,000 have already gone to Spain and Sinclair is looking for a similar demand from the French.

Alterations to the 'standard' QL include the provision of the AZERTY rather than QWERTY keyboard, French documentation and new versions of the four Psion packages.

An Italian version of the QL is due from Sinclair this month and the QL will no doubt be available in local language versions throughout Europe by the end of this year.

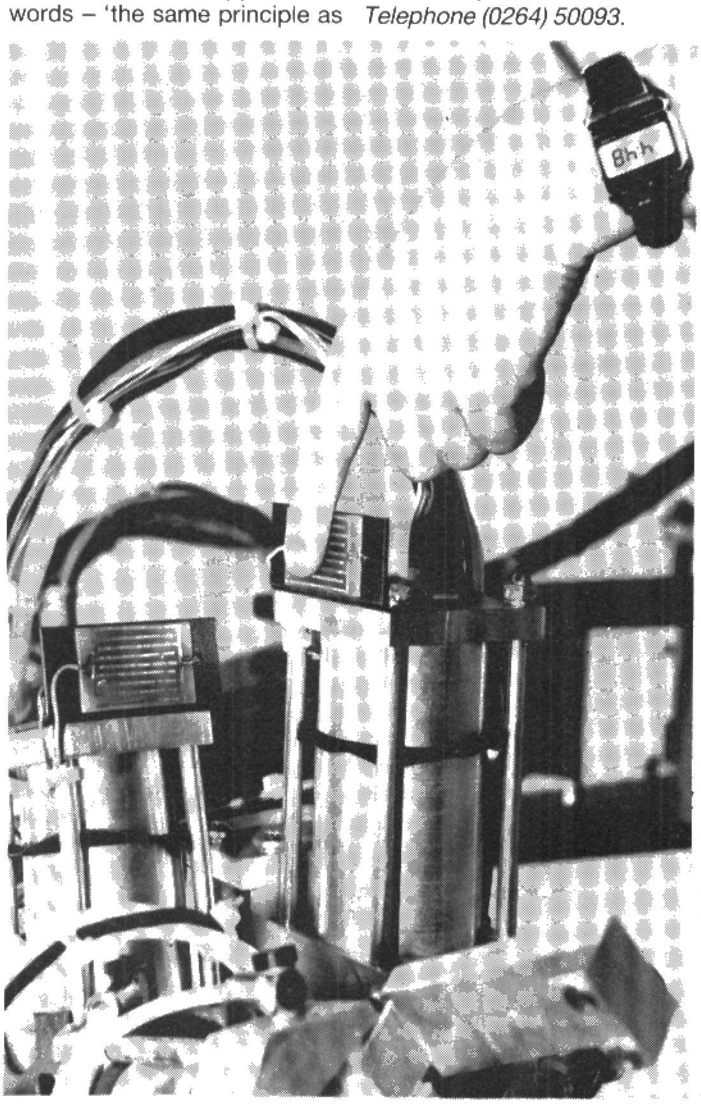
News about QL modems in this month's Comms news section and look out for an extensive review of modems in next month's magazine.



Full Prestel graphics and datacoms facilities are now available for Atari users – from Miracle Technology. See page 25 for full report.

Leading robots by the nose

Cybernetic Applications has added what are described as 'lead by the nose' sensors to their Neptune robot. The touch sensitive sensors enable the robot arm to be guided into position using – and again these are Cybernetic Applications words – 'the same principle as leading a bull by a ring through its nose'. The movements of the arm can be memorised by the control computer and repeated as many times as are required. *Cybernetic Applications, West Portway Industrial Estate, ANDOVER, Hants SP10 3LF. Telephone (0264) 50093.*



Watford's summer collection

Watford Electronics recently announced five new items of software.

The first is ROMSPELL, a spelling checker for use with both View and Wordwise. The ROM is accompanied by a dictionary disk containing 30,000 words.

ROMAS is an advanced assembler development system for the generation of assembly language programs for a range of eight bit micros, the BBC computer acting as the host processor. Processors supported include the 6502, Z80, 6809 and the Z8. ROMAS includes a full macro editor and cross assembler.

TRANSFERROM is a comprehensive utility package designed to transfer tape based software to specially formatted 'slave' disks. The easy to use software is driven via a series of menus.

The fourth item is a diagnostics disk, a menu driven package designed to exercise various sections of the BBC micro. Areas tested are the RAM, ROM, ULAs, Sound, Keyboard, Disk Drives, the 6502 and the Z80 second processor.

The final piece of software is Colour Art, which is used with the AMX mouse package and allows selection of up to 255 different colour shades.

Further details on all the packages from Watford Electronics at 250 High Street, WATFORD, Herts WD1 2AN.

Floppy disk stories

Xitan's Software newsletters often contain stories of the misfortunes that can befall floppy disks at the hands of users and, in the case of the following stories, the Post Office.

The first story concerns a company with 7½" letterbox and 8" disk drives. The postman managed to get a new batch of disks to the company by the simple expedient of folding over the last inch of the floppies.

But the best PO story must be the one about the Scottish postman who felt agrieved because one customer on his round didn't give him a Christmas tip. The next supply of disks was screwed up and forced through the letter box. Under the warning 'Floppy disks – Do not bend' was written 'Oh yes they do'.

If you have any floppy disk stories, or indeed any humerous computer stories, please write and let us know.

Beware

It has been brought to our attention that the Amateur Robot Association (ARA) has now ceased trading. The ARA have advertised in previous issues of *Electronics and Computing* and we would like to warn readers not to respond to any adverts that they may see in past issues of the magazines.

We would be interested to hear from any readers that sent the ARA any monies but have not do date received any response.

Data encryption

Worried about hackers interfering with your files? Paul Beverley's BBC software will encode any data sent from the RS423 port.

With the rapid growth of interest in hacking, I thought that, being on the side of right and truth and beauty, I should offer some help to those who want to protect their data. There are also doubtless many hackers who don't want to be hacked (think of the embarrassment) so here are some ideas on how to transmit data in coded form: in particular how to encode files prior to sending them and how to encrypt data as it is generated – so you can have a secure two-way communication in real-time.

To encode a text file for onward transmission through the mail by disk or tape requires a simple technique: you do an exclusive-OR (EOR) of each byte of data with a special key byte; this byte obviously has to be known by the recipient to enable him or her to reverse the process and read the file. But the EOR technique will only provide protection against non-technical snoopers. Anyone with a minimum of know-how could crack the code by trying each of the 256 possible key bytes in turn until the text made sense.

To make the code harder to crack you could use a four-byte key and then exclusive-OR each byte in the file with each of the four bytes in turn. The recipient uses exclusive-OR with the same bytes but in the reverse order. It sounds a good idea, but this code is simply unravelled. How? The quicker ones among you will realise what I did not when I originally arrived at this grand scheme: exclusive-OR'ing with four, or indeed any number of bytes can be shown to be equivalent to exclusive-OR'ing with one single byte and so it is in fact no improvement at all.

What you *should* do is take four bytes of the text at a time, and exclusive-OR them with the four byte key number. It is only necessary to prepare the text, using a word-processor if you wish, and save it on tape or disk as a file called "TEXT" and then run the following simple program:

```
10 K%=&AFBC9E3C
20 MODE0
30 MODE7
40 *LOAD TEXT 3000
50 FOR N%=&3000 TO &7000 STEP 4
60 IF !N%=0 E%=N%:N%=&7000
70 !N%=!N% EOR K%
80 NEXT
90 PRINT" *SAVE CODED 3000 ";E%
```

K% is the four byte key number, MODE 0 followed by MODE 7 clears an area of memory to all zeros and the text is then loaded into this memory (line 40). The FOR/NEXT loop is used to exclusive-OR the data, four bytes at a time, with the key number. Line 60 looks for the end of the file, as signified by the fact that four consecutive bytes are zero, and sets E% equal to the address of the end of the file. The final line prints out the line that needs to be copied from the keyboard in order to save the coded information back onto disk or tape. To decode a file, all you do is to repeat the process with the file names reversed – "CODED" at line 40 and "TEXT" at line 90 (or whatever you want to call the decoded file).

If you are a BASIC II user you can get the program itself to save the coded file without having to copy a line from the keyboard by using:

```
90 OSCLI("SAVE CODED 3000 "+
   STR$~E%)
```

TABLE 1

BIT 1	0	1	0	1
BIT 2	0	0	1	1
X OR SUM	0	1	1	0

LISTING 1

```
 10 HIMEM=&3000
 20 *RUN ENCRYPT
 30 *LOAD KEYDATA 3000
 40 ?&70=&30:REM start page number
 50 ?&71=15:REM No of pages
 60 REM Rest of program as original
 70
 80 MODE3
 90 PROCinitialise
100 REPEAT
110   PROCsend
120   PROCreceive
130   UNTIL0
140 END
150
160 DEF PROCsend
170 *FX3,5
180 *FX2,2
190 D%=INKEY0
200 IF D%>0 VDUD%
210 IF D%=13 VDU10
220 ENDPROC
230
240 DEF PROCreceive
250 *FX2,1
260 *FX3,0
270 D%=INKEY0
280 IF D%>0 VDUD%
290 ENDPROC
300
310 DEF PROCinitialise
320 ON ERROR PROCerror:END
330 VDU19;4;0;
340 *FX8,4
350 *FX7,4
360 ENDPROC
370
380 DEF PROCerror
390 *FX3,0
400 *FX2,0
410 REPORT
420 PRINT" at line ";ERL
430 ENDPROC
```

TABLE 2

ASCII 'A'	0100	0001
KEY	0101	0101
ENCRYPTED BYTE	0001	0100
ENCRYPTED BYTE	0001	0100
KEY	0101	0101
ASCII 'A'	0100	0001
ASCII 'A'	0100	0001
KEY 1	0101	0101
	1100	0100
KEY 2	1100	0011
	0000	0111
KEY 3	1010	1010
	1010	1101
KEY 4	0100	0010
ENCRYPTED DATA	1110	1111
ENCRYPTED DATA	1110	1111
KEY	1010	1110
ASCII 'A'	0100	0001

TABLE 3

ASCII 'ABCD'	0100	0001	0100	0010	0100	0011	0100	0100
FOUR BYTE KEY	1010	1111	1011	1100	1001	1110	0011	1100
ENCRYPTED STRING '_ _ yx'	1110	1110	1111	1110	1101	1101	0111	1000

_ represents unprintable character

Real-time encryption

The technique I have just described can only be used for sending pre-prepared files. You are not limited to sending them on disk or tape through the post; the files could just as easily be sent over an RS423 direct link or a modem link or by using the RS423/cassette link I described in the May issue. Nevertheless, the files still have to be prepared beforehand, so if you want to be able to "chat" to someone over a serial link and want the information to be secure, you need some form of real-time encryption.

Listing 2 allows you to have an extremely secure real-time communication using any of the RS423 or RS423/cassette links we have looked at in past articles (see the 'BBC networking' series). The program is a machine code routine working under interrupt, and every time a character passes through either the RS423 input buffer or the RS423 output buffer it is decoded or encoded automatically. To make the security really tight, the key is not a single byte or even four bytes but as much memory as you care to set aside for that purpose. Each byte that is sent is exclusive-ORed with one byte from the key-data, and an index is incremented so that the next byte to be sent is exclusive-ORed with the next byte in the key-data. At the receiving end, the same key-data is loaded into memory and each time a byte is received it is exclusive-ORed with the appropriate byte in the key-data. Clearly, both sending and receiving machines have to start at the same place in the key-data.

I have used separate indices for those characters which are sent and those which are received. If you are using a system which allows only one-way communication at a time, and if the input buffer is always emptied BEFORE any characters are put into the output buffer, then you could use a single index and thereby simplify the program, but as it stands, the program should work whatever software is used to set up the RS423 link.

Setting up the system

Listing 1 shows the modifications needed to the two-way RS423 communication program published in the April issue of *E&CM*. All you have to do is *RUN the machine code program, load in the key-data and set up two bytes to tell the encryption routines where the key-data is held in memory and how many (256 byte) pages of data are being used. The value of HYMEM has to be reduced to reserve this space, but if you are using only 1 page of key-data you could use page &B00 without changing HIMEM or, with a disk system, you could probably get away with using &1200 to &1900 provided you were not spooling any files down the RS423 link. It is important not to go right up into the page below the original value of HIMEM (&3F00 – &3FFF for MODE 3) as the top few bytes can get corrupted.

If a fixed position is used for the key-data you could include that information the set-

LISTING 2

```
   10 REM Program to add data encryption
   20 REM to an RS423 communications system
   30 REM (C)1985 Norwich Computer Services
   40 REM ********************************
   50 REM &70 to contain start page of key-data
   60 REM &71 to contain number of pages of key-data
   70 REM Also uses &72 to &78 inclusive.
   80 REM CALL &C00 to activate
   90
  100 PROCsetup_variables
  110 PROCassemble(&C00)
  120 PRINT'" *SAVE ENCRYPT ";~start" ";~P%" ";~start
  130 END
  140
  150 DEFPROCassemble(M%)
  160 FOR opt=0 TO 2 STEP 2
  170       P%=M%
  180       [OPTopt
  190
  200 .start
  210       SEI              \ Reset vector for
  220       LDA #newremVEC MOD 256
  230       STA remVEC       \ removing characters
  240       LDA #newremVEC DIV 256
  250       STA remVEC + 1  \from buffers.
  260       CLI
  270
  280       LDA #0           \ Initialise variables
  290       STA Rpoint       \ Low bytes of
  300       STA Spoint       \ addresses.
  310       LDA startPAGE
  320       STA Rpoint+1     \ High bytes of
  330       STA Spoint+1     \ addresses.
  340       LDA pages
  350       STA Rcount       \ Set counters
  360       STA Scount       \ for number of pages.
  370       RTS
  380
  390 .newremVEC
  400       BVC OK           \ V set if not a "remove"
  410
  420 .doit
  430       JMP oldremVEC    \ Go to OS routine.
  440
  450 .OK
  460       CPX #1           \ If not buffer number 1
  470       BNE RS423out     \ it might be RS423 o/p buffer.
  480
  490 .RS423in
  500       JSR oldremVEC    \ go and get the character,
  510       BCS rts          \ but give up if there wasn't one.
  520
  530       PHP              \ Preserve flag status
  540       TYA              \ Put character in accumulator.
  550       LDY #0           \ EOR with the byte pointed to
  560       EOR (Rpoint),Y   \ by Rpoint.
  570       INC Rpoint       \ Point to next key-data byte.
  580       BNE out          \ If at the end of a page,
  590
  600       INC Rpoint+1     \ increment page number
  610       DEC Rcount       \ and count down.
  620       BNE out          \ If last page in key-data,
  630
  640       LDY pages        \ reset pages count, and
  650       STY Rcount
  660       LDY startPAGE    \ reset the start address
  670       STY Rpoint+1     \ of the key-data.
  680
  690 .out
  700       TAY              \ Put byte back into Y.
  710       PLP              \ Restore flag status.
  720
  730 .rts
```

ting up routines in the machine code program by using, for example:

```
361  LDA #&30
362  STA &70
363  LDA #15
364  STA &71
```

These are the equivalents of lines 40 and 50 in **Listing 1.** Doing it that way means that you should be able to use this encryption program even with ROM-based data transfer programs, always assuming they do not clash over the use of memory locations. The two pointer addresses stored in &73/4 and &75/6 must be zero page locations, but the other variables could be stored anywhere in memory, not necessarily in zero pages.

Loading the key data can be done either from the keyboard or from the program as illustrated in **Listing 1;** as can the loading and running of the machine code program. Disk users can just use *ENCRYPT instead of *RUN ENCRYPT. To generate the key'data in the first place you could use:

```
FOR N%=&3000 TO &3F00 STEP 4:
!N%=RND:NEXT
*SAVE KEYDATA 3000 3EFF
```

Machine code programmers should be able to see how the routines work by looking at the annotations within the program itself. The basic idea is that REMVEC is intercepted, ie every time the OS tries to remove a character from either of the RS423 buffers, this routine does a JSR to the real REMVEC routine and then, if no character is available, gives up. If however there is a character available, the byte is EORed with the byte pointed to by the index, and the index incremented, before returning control to the operating system.

```
 740        RTS
 750
 760 .RS423out
 770        CPX #2          \ If not buffer 2
 780        BNE doit        \ then it wasn't RS423 output.
 790
 800        JSR oldremVEC   \ If it was, do it
 810        BCS rts         \ in exactly the same way
 820
 830        PHP             \ as for the RS423 input routine,
 840        TYA             \ using different pointers,
 850        LDY #0          \ but the same key-data.
 860        EOR (Spoint),Y
 870        INC Spoint
 880        BNE out
 890
 900        INC Spoint+1
 910        DEC Scount
 920        BNE out
 930
 940        LDY pages
 950        STY Scount
 960        LDY startPAGE
 970        STY Spoint+1
 980        TAY
 990        PLP
1000        RTS
1010
1020        ]
1030        NEXT
1040 ENDPROC
1050
1060 DEFPROCsetup_variables
1070 remVEC=&22C
1080 oldremVEC=!remVEC AND &FFFF
1090 OSBYTE=&FFF4
1100 startPAGE=&70
1110 pages=&71
1120 Ystore=&72
1130 Rpoint=&73 : REM must be in zero page
1140 Spoint=&75 : REM must be in zero page
1150 Rcount=&77
1160 Scount=&78
1170 ENDPROC
```

Multipurpose Centronics interface

Most computers provide a Centronics printer port. Plug this interface board into it and you can control up to four items of equipment.

Computer control of equipment opens up a vast range of possibilities – from intelligent control of the heating system to automatically closing the curtains at dusk. To assert such control of electrical equipment will require that you equip your computer with some form of interface.

Our Centronics interface is designed to plug into the printer port of a micro computer and provides a simple-to-program unit that can switch up to four items of equipment. It also features four LEDs to indicate the status of the system.

In the rapidly changing world of computing, the specification of the Centronics parallel interface is one of the few reliable standards. The general purpose interface that forms the basis of this article is designed for use in conjunction with the printer port of a microcomputer and provides up to four relay switched outputs and four status indicators.

We took the decision to use the Centronics port in order to achieve a design that could be used on a wide range of hardware. Some alternative approaches considered included the standard techniques of memory mapping and I/O mapping. The former involves making the interface appear as a standard memory location, but data written to this 'special' location is decoded and used to control the interface. The second technique makes use of the I/O commands built into many micros to control the interface.

Both the above techniques have two serious drawbacks. The first is that direct access to the computer's signal bus is required in order that extra decoding circuitry can be installed. The second is that specialised driving programs are required control the interface.

By designing the interface to operate with a Centronics interface, most, if not all, of these problems are overcome.

Basic operation

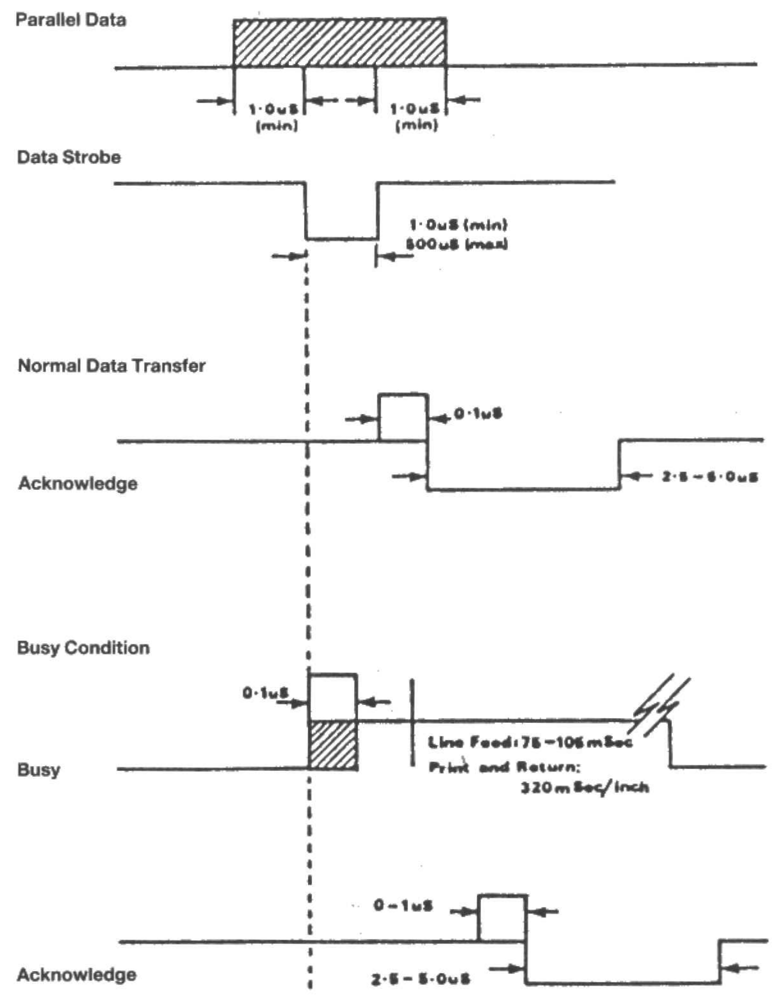
The timing and signal relationships of the interface are shown in **Figure 1.** The sequence of events is as follows: the computer first places the data to be transmitted to the printer/interface on the pins of the interface connector; it then strobes the line labled data strobe low, indicating to the device connected to the printer interface connector that the computer is ready to send a new valid data word.

The device connected to the port signals that it has successfully received the data by strobing the line marked 'acknowledge' low.

LISTING 1

```
10 A%=240              : REM ALL
                         RELAYS OFF
20 A%=A%+1+4           : REM RELAYS
                         1 AND 3 ON
30 A%=A%-16-64         : REM LEDs 1
                         AND 3 ON
40 LPRINT CHR(A%)      : REM ACTIVATE
                         THE
                         INTERFACE
50 FOR I=1 TO 100      : REM DELAY
60 NEXT I              : REM DELAY
70 A%=A%-1+16          : REM RELAY 1
                         OFF, LED 1 OFF
80 LPRINT CHR(A%)      : REM ACTIVATE
                         THE
                         INTERFACE
90 END
```

Figure 1. Timing and signal relationships.



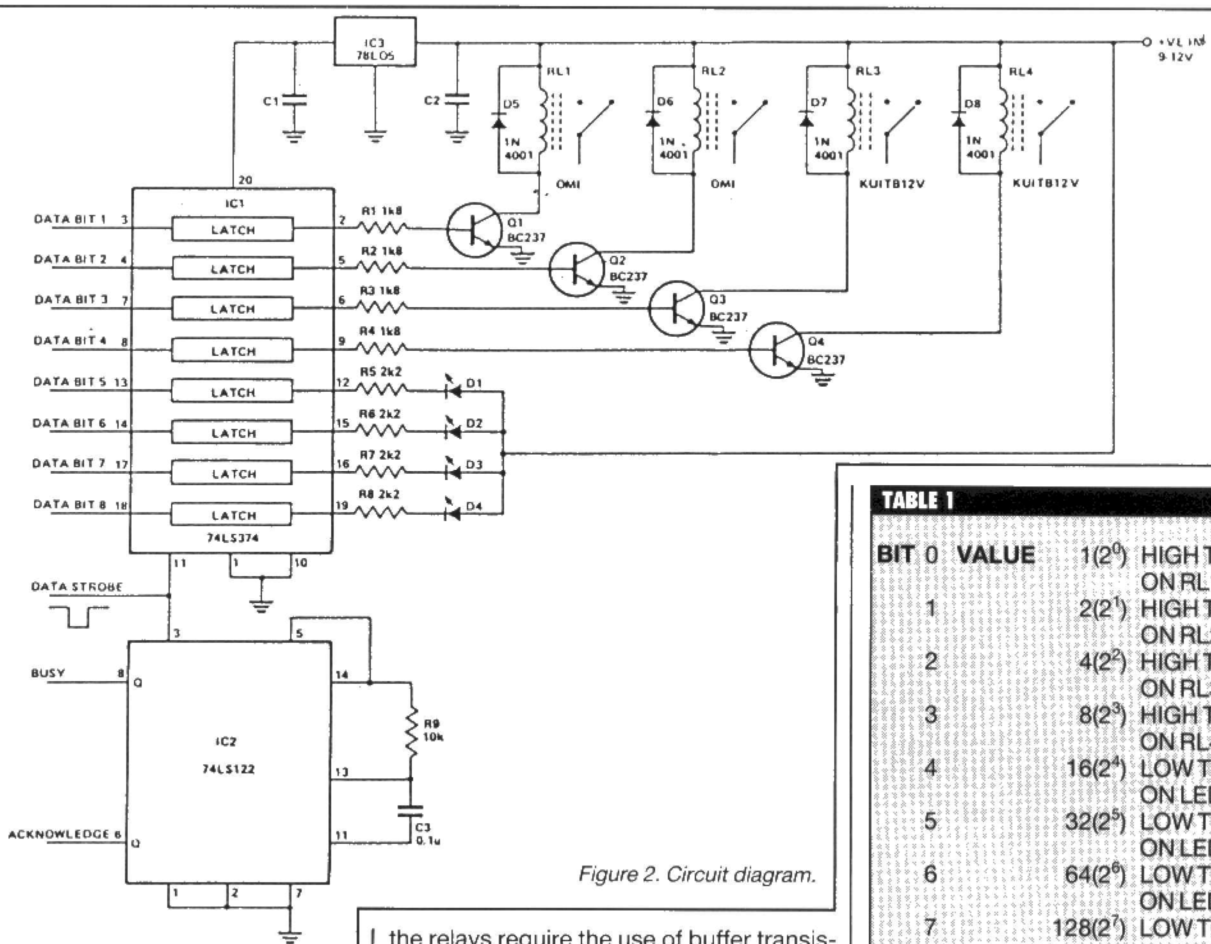


Figure 2. Circuit diagram.

The third control line that is of interest to us is marked 'busy'. The external device can indicate to the computer that it is temporarily unable to accept data by taking this line low.

As can be seen from the timing diagram the relationship between the acknowledge and busy lines is such that any interface need only monitor one of them to operate successfully.

Circuit description

The major elements of the interface circuit are a data latch to grab the data and hold on to it and a monostable to feed back the acknowledge and busy signals.

Thus every time the computer strobes data-strobe low, the latch, IC1, latches the data on its input pins. The data will now appear on the output lines of the latch irrespective of what happens to the data on its input lines.

The rising edge of the data strobe signal triggers the monostable, IC2, which generates the busy and acknowledge signal that is fed back to the computer.

IC3 is a 5V regulator that provides a stabilised supply for the logic gates from a higher, unregulated supply.

The interface has two sets of outputs – relay and LED. The relay outputs are for switching external equipment and the relays for prompts etc. The LEDs are driven directly from the outputs of the latch, while the relays require the use of buffer transistors. The diodes across the relays are to protect the transistors from the back EMFs generated as the relays switch.

Construction

Construction of the project requires little in the way of comment. All the smaller components should be mounted first, leaving the ICs (fitted in sockets for preference) and the relays until last.

When construction is complete, before applying any power to the board, double check that there are no solder bridges evident and that all components are in the correct position.

A 12V power supply should then be connected to the board – it should draw no more than 200mA. If all is in order, the interface can now be connected to your computer.

In use

Each line of the interface is driven from a separate Centronics data line as shown in **Table 1.**

A logic '1' or 'high' on a data line will energise the corresponding relay, operating its contacts. Thus, for example, to switch on relay 1 it is only necessary to output the character whose value is 1 to the interface. In Microsoft BASIC this would be accomplished with the statement:

LPRINT CHR$(1);

Note the use of the semi-colon, this is to suppress the output of carriage return line-feed, which would otherwise output values of 10 and 13 to the interface, immediately overwriting the 1 originally intended.

TABLE 1

BIT	VALUE		
0		$1(2^0)$	HIGH TO TURN ON RL1
1		$2(2^1)$	HIGH TO TURN ON RL2
2		$4(2^2)$	HIGH TO TURN ON RL3
3		$8(2^3)$	HIGH TO TURN ON RL4
4		$16(2^4)$	LOW TO TURN ON LED 1
5		$32(2^5)$	LOW TO TURN ON LED 2
6		$64(2^6)$	LOW TO TURN ON LED 3
7		$128(2^7)$	LOW TO TURN ON LED 4

Control of the LEDs is slightly different. As with the relays each LED is driven from a separate data line. A logic low on a data line causes the LED to be switched on, whereas the reverse is the case for the relay control lines. To reset the interface to a condition where all LEDs and relays are off requires that the binary word '11110000' is sent. This is decimal 240. To turn on any one LED, the value shown in **Table 1** should be subtracted from this value.

It is possible to control the action of the relays and the LEDs simultaneously. In order to accomplish this, simply add together the values given in Table 1 for each of the relays to be switched. For example to turn on RL1, RL3 and LED3 it would be necessary to alter the current interface control number with the following string:

+1+4–64

The short program shown in **Listing 1** illustrates the use of the interface. It will first turn on relays 1 and 3, turn on LEDs 1 and 3, and turn off relay 1 and LED 1 after a short delay.

The ease of use and general purpose nature of this interface make it an ideal introduction toe the world of computer control techniques.

A full kit of parts for this project can be obtained from Cirkit Distribution, Park Lane, Broxbourne, Herts EN10 7NQ. Price £22.50 plus VAT and 60p p&p.

NET NEWS

New modems end QL isolation

By the time you read this, the QL's long and lonely sojourn outside the comms world should have ended.

At least three QL modems are on the eve of delivery, among them the Sinclair-backed QCOM device designed by troubled OE Ltd of Cumbria, as well as units from Medic Data Systems and Modern House.

The QCOM system, comprising stackable controller, modem and auto-dial units, and featured in the recent Sinclair advertising, has been bought by established comms company Tandata from the OE receiver, and looks set for early manufacture at AB Electronics in South Wales.

We've had an extensive play with the device (which will cost around £200) and can report, not only a full range of comms facilities, but also a high degree of integration with the computer itself – allowing pre-programmed activities to be carried out automatically on data received from a remote source.

Medic's device, meanwhile, forms part of the company's integrated system of QL add-ons, and thereby requires modem buyers to invest in a £99 motherboard – pushing the price to £250.

But the modem looks impressive enough, employing the latest 7911 chip to supply an excellent range of features, including auto-dial and answer, auto-baud rate selection, and European and Bell formats.

As a curious bonus, the company promises early availability of software to turn the device into an answering machine.

The £200 Modem House unit, which we've also laid hands on, offers similar facilities with a similar chip – the 7910. Called Bright Star, the modem was designed by Commpak, and has the added feature of a Centronics port with integrated printer and modem functions – so that real time screen dumps, using the 2K buffer if necessary, become a possibility.

***Tandata** 06845 68421*
***Medic Data** 0256 475244*
***Modem House** 0392 69295*

Apricot goes on-line

ACT has launched a private viewdata system, using the Apricot Xi10S Micro, which provides up to 8,000 pages, supports 16 simultaneous lines and gives access to 200 remote terminals.

Apricot Viewdata is basically a mini-Prestel system (it's totally compatible with Prestel), and much cheaper than currently available minicomputer based systems.

Communication is through the public telephone network, packet switch stream (PSS) or internal telephone exchange.

Like Prestel, it features two levels of passwords and part of the database can be restricted by putting users in a Closed User Group (CUG).

The system is aimed at businesses requiring their own in-company information service, and heavily undercuts the £100,000 cost of comparable systems.

For £9,930, the set-up gives you the Xi10S with 512K RAM and 10Mb hard disk, an ACT colour card, colour monitor, viewdata software and AMAC – ACT's own modem multiplexer.

Data coms for Atari users

Full Prestel is now available to users of the Atari range of micros (including the new 130XE) thanks to modem manufacturers Miracle Technology and their £59.95 Multi-Viewterm/Datatari modem interface and software package.

The package handles baud rates of 300/300, 1200/75, 75/1200 and 1200/1200, and includes a cable with an Atari 13-way peripheral port plug at one end and a 25-way plug at the other, suitable (of course!) for Miracle's own range of modems, including the acclaimed WS2000.

The Datatari interface and Multi Viewterm disk based software also gives access to Electronic mail, Telex, database and user to user communications and telesoftware downloading.

You can get it through Atari main dealers, including the Silica Shop, or by mail order from Miracle Technology, St Peter's Street, Ipswich IP1 1XB.

Do-it-yourself on Micronet

Micronet's gallery service, allowing members to edit their own Prestel pages, is now operational.

For 25p, Micronet subscribers can have complete editorial control (well . . . almost) of one Prestel frame for 6 months. Changes to the frame, carried out automatically by the system through a once-daily sweep of special response frames, will cost the user 4p a time. And access to the frame will be via an automatically-generated index of user names.

Each frame will carry a subscriber's name and account number, allowing 16 lines for their literary or artistic masterpieces.

Meanwhile, Micronet's Celebrity Chatline feature continues to flourish. Recent guests to receive a live grilling by subscribers have included Richard Hooper – the BT boss of value added networks – and Acorn Chairman Dr Alex Reid.

A Micronet member has done something to redress the service's image as a hot-bed of hacking naughtiness. Joe Rice has persuaded the National Research Council to back development on his vein-recognition device for database security.

Joe believes that his system, by which a bar-code reader – similar to those employed in supermarkets – would be used to scan the pattern of blood vessels in the wrist against a store of accredited users, provides the first ever clean and fool-proof remote recognition device.

Prestel standard extends to Far East

The Prestel standard for viewdata transmission has received another boost in its seemingly inevitable course towards world domination with news of an order from Singapore Telecoms for £15m of GEC equipment.

The British equipment will provide Singapore with a unique joint viewdata and teletext system, called Teleview, by which users will send their request for information via the phone link, but receive the data as television signals – so that high resolution images, essential for displaying the Chinese character set, can be easily transmitted.

The broadcast data will have to carry identifiers that enable individual TV sets to be addressed. As a result, and in order to cope with a potential one million users, an entire channel is being set aside for the service.

As well as the viewdata and teletext service, Teleview will also provide two-way telex, full colour photograph-quality display, electronic messaging, and dynamically re-definable character sets.

If trials are successful, the initial delivery of 1100 terminals may be followed by orders worth over £1 billion.

NET QUESTIONS

Q: I know that to transmit data over a two wire system, such as the public telephone network, requires that data be presented in a serial form. Such a data stream is available at the RS423 port of my BBC micro, so why can't it be connected directly to the telephone system bypassing the need for a modem?
N. McKechnie.

A: While it is tempting to think of the PSTN (Public Switched Telephone Network) as a means by which direct two wire links can be initiated between any two 'phones in the system, the reality is far removed from this simple concept. The thing to remember is that the network was designed for the transmission of the human voice and has been optimised to carry out this task with the greatest efficiency.

The data output from the RS423 port of the BBC micro is very different from the signal produced by human speech. First, it is – in the case of commonly used transmission rates – quite a low frequency. Secondly, when compared to a typical speech it is of a very large amplitude. Attempting to pass the RS423 signal down the telephone line would play havoc with BT's line amplifiers, filters, switching equipment and would probably put your exchange line out of action.

The main function of a modem's hardware is to translate the digital RS423 signal into something that resembles the human voice as far as the PSTN is concerned. It does this by allocating specific audio frequencies to the two (ON and OFF) digital states. The frequencies chosen fall within the range in which most of the energy of human speech is contained. The frequencies are very strictly controlled. BT uses similar audio tones at different frequencies to remotely control various items of equipment and it is important that a modem's output does not interfere with these functions.

Q: I have been given an ex GPO Type 13A modem which is marked 300 baud FULL DUPLEX (originate only) and have had it demonstrated on a BBC B using the Maplin CASHTEL and the software from their catalogue. The BBC has RS423 which is I understand +5V.

I wish to use the modem with my QL computer which has RS232C (+12V) but all I get is rubbish with just enough to show me that it was indeed Maplin's computer.

Please can you tell me why?
E. Duncan-Dunlop.

A: First of all may we say that there is insufficient detail in your letter for us to provide a complete reply. There are no details of the connections you have made between the modem and the QL, nor do you say what QL terminal emulation software is being used. In order to configure a working comms system a multitude of aspects concerning both hardware and software have to be correctly specified. Many commercial companies have spent many months developing dedicated systems for particular computers and if you wish to get your modem working with the QL you will have to do a lot of research into comms protocols and the operation of the hardware of your modem. Once you have done this it could still be that, unless you have a good degree of technical knowledge that you will fail to get the system working correctly.

It would probably be better for you to purchase one of the dedicated modems that have started to appear for the QL. Our news pages this month give details of some of these products. In addition, by purchasing a modem with a 1200/75 baud capability, you would be able to use you QL to communicate with services operating to Prestel protocols. The Type 13A modem is not compatible with the transmission rates used by these services.

SPECIAL ANNOUNCEMENT

Electronics and Computing is changing

The publication of our October issue on September 13th will mark another stage in the evolution of *Electronics and Computing*.

As we near our sixth year of publication we've decided to make some changes to the presentation of *E&CM*.

We'll be expanding our coverage of the microcomputer industry with particular reference to the new generation of 16 bit computers and the expanding world of communications. Regular readers will also find their established favourites in the new look magazine.

More details next month . . . in the meantime remember . . .

— The age of computing is nigh

NET NUMBERS

This is a list of all known (legal!) databases available. If you know of any more, let us know. The list is split into sections, grouped under the baud rates at which they operate. Just look at the list which corresponds to the type of modem that you have, and get dialling!

Note: Protocol is shown as number of bits per word, parity, number of stop bits. Bits per word can be 7 or 8, parity can be none (n), even (e) or odd (o) and stop bits can be 1 or 2. See your modem's instruction manual for details.

300 Baud Systems

Name: **BABBS**
Protocol: 8,n,1
Phone: 0225 23276
Times: Weekdays 9pm-midnight
Weekdays 9pm-9am
Notes: Atari based

Name: **BABBS 1**
Protocol: 8,n,1
Phone: 0394 276306
Times: 24hrs
Notes: Runs on an Apple but caters for everyone

Name: **BABBS 2**
Protocol: 8,n,1
Phone: 0269 778956
Times: 24hrs
Notes: Apple based

Name: **BASUG**
Protocol: 8,n,1
Phone: 0742 667983
Times: 24hrs day
Notes: British Apple Systems User Group

Name: **Bettisfield**
Protocol: 8,n,1
Phone: 094875 378
Times: 9am-9pm daily
Notes: Remote CP/M system

Name: **BITEC**
Protocol: 8,n,1
Phone: 0268 25122
Times: 24hrs

Name: **Birmingham North**
Protocol: 8,n,1
Phone: 0827 288810
Times: 24hrs
Notes: Runs on a Tandy. Strong MUD SIG

Name: **Bloxam**
Protocol: 8,n,1
Phone: 0295 720812
Times: Weekdays 7am-8am, 6pm-7pm
Weekends 2pm-6pm

Name: **CBBS-NE**
Protocol: 8,n,1
Phone: 0207 543555
Times: 2.30pm-9am daily
Notes: Mainly business users

Name: **CBBS Southwest**
Protocol: 8,n,1
Phone: 0392 53116
Times: 24hrs
Notes: Also runs at 1200/75 baud

Name: **CBBS Surrey**
Protocol: 8,n,1
Phone: 04862 25174
Times: 24hrs

Name: **Chiltern**
Protocol: 8,n,1
Phone: 07073 28723
Times: 9pm-8am daily

Name: **City BB**
Protocol: 8,n,1
Phone: 01 606 4194
Times: 24hrs
Notes: Also runs at 1200/75 baud

Name: **CNOL**
Protocol: 8,n,1
Phone: 0524 60399
Times: 24hrs
Notes: Mainly medical information

Name: **Communitree Hope**
Protocol: 8,n,1
Phone: 0874 711147
Times: 24hrs
Notes: Tree-structured. Very friendly. Caters for Welsh language.

Name: **Distel**
Protocol: 7,e,1
Phone: 01 679 1888 or 679 6183
Times: 24hrs
Notes: Run by Display Electronics. Remote ordering system.

Name: **Forum 80 Hull**
Protocol: 8,n,1
Phone: 0482 859161
Times: Tue/Thu 7pm-10pm
Weekends 1pm-10pm
Notes: The original BB, as run by Fred Brown.

Name: **Forum-80 Spa**
Protocol: 8,n,1
Phone: 0926 39871
Times: 1pm-midnight
Notes: Lots of info for Tandy users

Name: **Hamnet**
Protocol: 8,n,1
Phone: 0482 497150
Times: 24hrs

Name: **Livingstone BBS**
Protocol: 8,n,1
Phone: 0506 38526
Times: 24hrs

Name: **London Underground**
Protocol: 8,n,1
Phone: 01 863 0198
Times: 7pm-midnight daily

Name: **Mailbox-80 Warley**
Protocol: 8,n,1
Phone: 0384 635336
Times: Mon-Sat 5.30pm-8.30am
All day Sunday

Name: **Mailbox 80 Liverpool**
Protocol: 8,n,1
Phone: 051 428 8924
Times: 24hrs
Notes: Run by Peter "PCW" Toothill

Name: **Maptel**
Protocol: 8,n,1
Phone: 0702 552941
Times: 24hrs
Notes: Online ordering service for Maplin Electronics

Name: **MBBS**
Protocol: 8,n,1
Phone: 01 640 2617
Times: Thu-Sun 10am-10pm

Name: **Micro Live**
Protocol: 8,n,1
Phone: 01 579 2288
Times: 24hrs
Notes: Operated by the BBC Micro Live team. Lots of useful notes.

Name: **Microweb**
Protocol: 8,n,1
Phone: 061 456 4157
Times: 24hrs

Name: **MOBBS**
Protocol: 8,n,1
Phone: 061 736 8449
Times: 24hrs
Notes: Also available at 1200/75 baud

Name: **OBBS**
Protocol: 8,n,1
Phone: 061 427 1596
Times: 24hrs

Name: **NBBS Chingford East**
Protocol: 8,n,1
Phone: 0692 630186
Times: Sun-Thu 10pm-7am
Fri-Sat 10pm-10am

Name: **PIP**
Protocol: 8,n,1
Phone: 0742 667983
Times: 24hrs
Notes: Operates on US Bell tones from midnight to 8am

Name: **Southern BB**
Protocol: 8,n,1
Phone: 0243 511077
Times: 24hrs

Name: **SITEC**
Protocol: 8,n,1
Phone: 0782 265078
Times: 24hrs
Notes: Stocke ITeC. Remote CP/M system.

Name: **TBBS London**
Protocol: 8,n,1
Phone: 01 348 9400
Times: 24hrs

Name: **TBBS Metro**
Protocol: 8,n,1
Phone: 01 341 7840
Times: 24hrs
Notes: Also runs at 1200/75 baud

Name: **TBBS Blandford**
Protocol: 8,n,1
Phone: 0258 54494
Times: 24hrs

Name: **WABBS**
Protocol: 8,n,1
Phone: 0903 42013
Times: 24hrs
Notes: Atari-based system

1200/75 baud systems

Note: Protocols for these systems is 7 bits, even parity, one stop bit unless otherwise stated. Format is either normal, scrolling screen display or Prestel-type viewdata display.

Name: **Aberdeen ITeC**
Format: Prestel
Phone: 0224 641585
Times: 24hrs
Notes: CommunITel viewdata system

Name: **Bildschirmtext**
Format: Prestel
Phone: 010 4930 1511
Times: 24hrs
Notes: Germany's Videotext system. Calls are EXPENSIVE!

Name: **Bulletin**
Format: Prestel
Phone: 0462 677177
Times: 24hrs
Notes: Public system run in ICL mainframe by Herts Council

Name: **CBBS Southwest**
Format: Scrolling
Phone: 0387 53116
Times: 24hrs
Notes: Public system run in ICL mainframe by Herts Council

Name: **CBBS Southwest**
Format: Scrolling
Phone: 0392 53116
Times: 24hrs
Notes: Also runs on 300 baud, 8,n,1

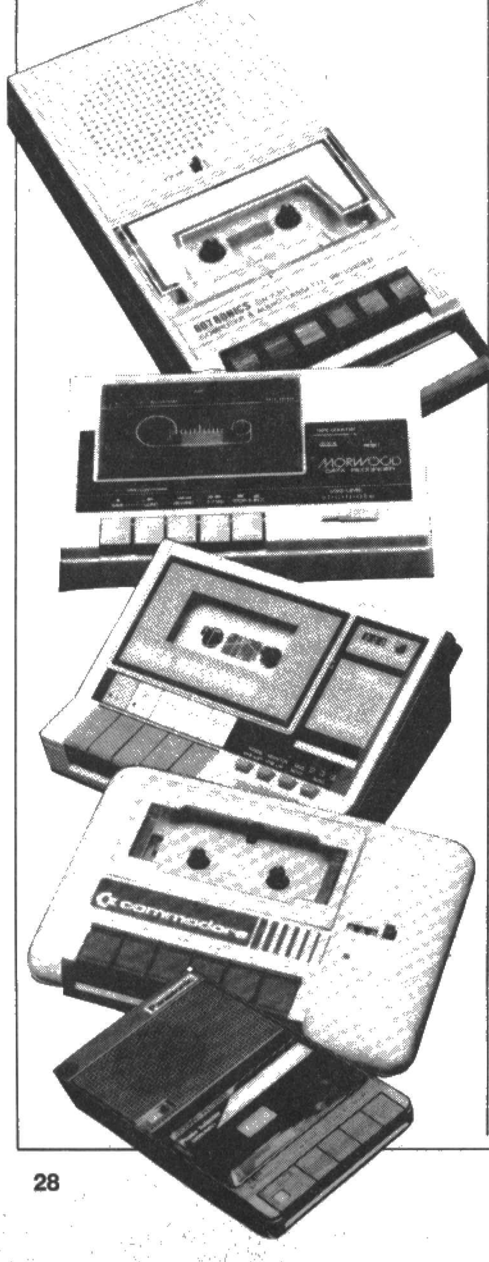
Name: **City BB**
Format: Scrolling
Phone: 01 606 4194
Times: 24hrs
Notes: Also on 300 baud, 8,n,1

Name: **C-View**
Format: Prestel
Phone: 0702 546373
Times: 24hrs
Notes: Run by Rochford County Council

Name: **Distel**
Format: Scrolling
Phone: 01 679 6183
Times: 24hrs
Notes: Also on 300 baud, 8,n,1

Getting it taped

Mike James continues his series on mass storage techniques with everything you need to know about taping – and how to keep your head straight.



Magnetic tape storage must rank as one of the earliest and most popular types of secondary storage.

Mag tape spans the range from the most expensive to the cheapest form of storage available. And, whereas for large mainframe computers magnetic tape storage implies the use of equally large tape drives and ½" tape wound on 12" reels, it is difficult to see how personal computers could ever have caught on without the use of the small, cheap, audio cassette. In this series the emphasis falls naturally on storage devices suitable for personal computers, so most of this article describes how audio cassette storage works and how it can be made reliable – but it is worth taking a look over the fence to see what direct digital recording has to offer.

Recording on tape

The physical principles involved in recording on tape are known to anyone who has magnetised a piece of Iron using a bar magnet. A magnetic field will magnetise some materials which will then retain their magnetism for a long time. Magnetic tape is made by coating a strong plastic tape with particles of iron oxide. The tape can be thought of as made up of a large number of 'domains' each like a small bar magnet. In an unmagnetised tape the domains are in an unordered state and their magnetic fields tend to cancel one another out, giving an overall magnetisation of zero – see **Figure 1.** However if a portion of the tape is subjected to a magnetic field this tends to make the domains line up in one direction and this gives the tape a net magnetic field – see **Figure 2.** The number of domains that line up depends on the strength of the magnetic field to get out except at the gap and hence so does the degree of magnetisation.

In a tape recorder the tape is magnetised by passing in front of a recording head – an electromagnet – which produces a varying field in response to the current flowing in it – see **Figure 3.** The design of the head is crucial to the performance of the tape recorder. The head is a ring of magnetic material with a single gap that faces the tape. The reason for this strange design is simply that the ring doesn't allow the magnetic field to get out except at the gap (see **Figure 3**). So it is possible to create a very localised magnetic field that can influence the magnetisation of a very small part of the tape at a time. In practice the gap in the ring is filled with some non magnetic material so if you examine most recording heads all you will see is a dark vertical line. Obviously to produce a pattern of magnetisation on the tape that corresponds to the pattern of current in the head, the tape must be moved on before the current in the head can change. How much it must be moved depends on the area of the tape that the head affects at any moment – and this depends on the size of the head gap. In other words the maximum frequency that you can record on a tape depends on:

1) the size of the head gap – the smaller the higher the frequency
2) the speed that the tape moves – the faster the higher the frequency.

Also – as the claims and counter-claims of the manufacturers indicate – the maximum frequency also depends on the nature of the tape itself. The smallest change in magnetisation that the tape can register depends on the size of the magnetic domains, ie. the material the tape is made of – and how it was made – is clearly important. In practice the maximum frequency is governed more by the size of the head gap and the speed of the tape but in later articles we will take a look at some ways to improve the basic tape medium so that it can record fantastic amounts of data.

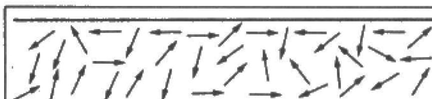


Figure 1. The unordered state of the domains in an unmagnetised tape.

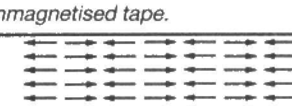


Figure 2. The domains line up when a tape is magnetised.

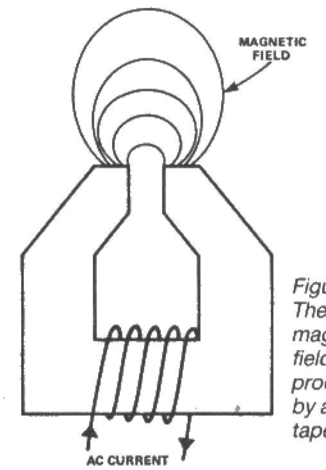


Figure 3. The magnetic field produced by a typical tape head.

Until recently most small computer tape decks (cassette and ¼" cartridge) were limited to around 2Mbytes of storage per tape, because data had to be recorded in bursts so that the tape contained blocks of data with a length of blank tape between. The blank sections of tape were to allow time for the tape to start and stop in between reading and writing data. In practice a tape would usually contain more blanks than usefully recorded data. A very special type of tape has now been developed to overcome this problem – the 'streaming tape'. In a streaming tape the data is recorded in one go and so there is no need for the start/stop gaps. Streaming tapes achieve storage capacities of up to 60Mbytes per tape but of course all this data has to be written and read in one operation! You might find it difficult to see why anyone would want a streaming tape if you have never considered the problem of making backup copies of 10Mbyte and 20Mbyte hard disks – a streaming tape will store and restore an entire disk in one operation.

It is unfortunate that digital cassette decks have never really managed to fall in price as a result of the personal computer revolution – they still cost over £500 per unit. If some enterprising manufacturer had come up with a cheap, reliable digital deck I'm sure we could have avoided most of the problems involved in using audio decks.

Using audio decks for data

Standard audio cassette decks have been used for data recording almost since the very start of personal computing. If you want to store binary data using an analogue recorder then the most obvious way of doing it is to use some sort of tone code. One of the simplest and most widely used is to associate a binary one with one frequency and zero with another – this is called a 'frequency shift code'. Back in the early days of personal computing a standard for this type of data recording was set. The Kansas City standard, or CUTS (Computer Users Tape Standard) as it was called, used eight cycles of a 2400Hz signal to represent a one, and four cycles of 1200Hz to represent a zero. The frequencies were chosen because they were easy to obtain from most computer systems and, because one is double the other, a 'clock' signal could be derived from the recording – the clock signal made it possible to correct for speed variations. Both the BBC Micro and the Dragon (to name but two) use this standard to an extent, but as the standard doesn't specify the format in which data should be stored on tape – just the codes for zero and one – don't expect to be able to transfer tapes between the BBC and the Dragon!

The trouble with the Kansas City standard is that it is a little wasteful of tape; for example it records eight cycles of tone for each bit, so setting the data rate at 30 bytes a second. Some machines, notably the BBC Micro have managed to modify the standard to work at 120 bytes per second.

There are many alternatives to the Kansas City standard in use today. The Spectrum records data using a single long pulse for a one and a single short pulse for a zero. (For a full description of the Spectrum's tape system see *An Expert Guide to the Spectrum* by Mike James, published by Granada 1984.) The data is recovered by measuring the time between the rising and falling edges on the replayed signal from the tape recorder. Speed insensitivity is obtained by allowing a wide range of times to define a long and short pulse.

Bias and erase

Playback is achieved simply by moving the tape past a similar head, and in low cost recorders the same head that was used to record the tape. The magnetic variations induce a small current to flow in the coils which are then amplified and corrected for any distortions (and if we used the simple recording scheme outlined above there would certainly be a lot of distortion!)

It takes a certain minimum field before any of the magnetic domains will begin to align themselves. In other words, small signals simply do not record and in general waveforms are very distorted (see **Figure 4**). The solution is to apply a 'bias' signal that lifts the signal clear of the 'dead zone'. In expensive recorders the bias takes the form of a high frequency ultrasonic tone that the audio signal modulates **(Figure 5)**. In cheaper recorders the bias is a DC current but this reduces the signal-to-noise ratio of the system.

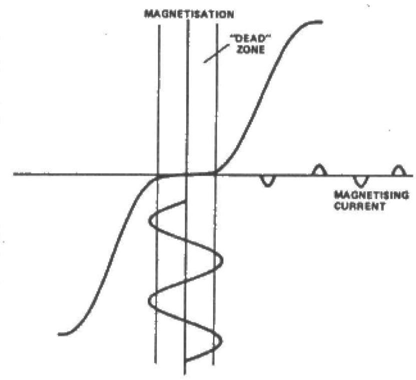


Figure 4. Without bias a great deal of distortion is the result.

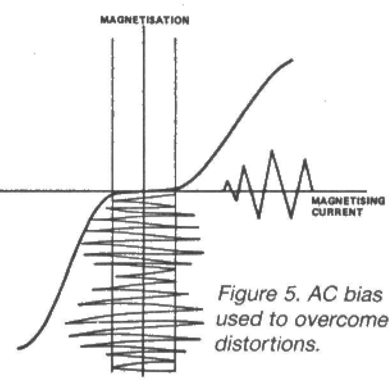
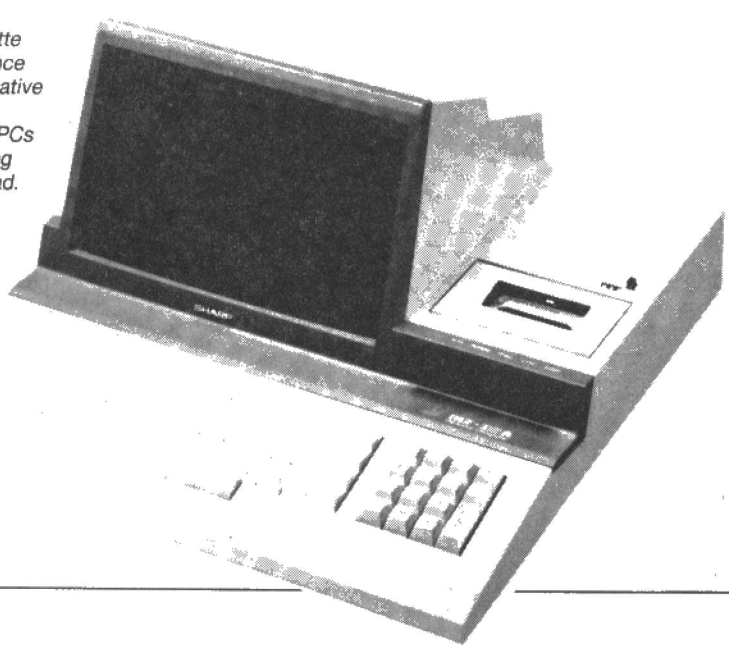


Figure 5. AC bias used to overcome distortions.

Even with the use of bias there is considerable distortion because the frequency response of the system is far from flat. To compensate for this most recorders boost frequencies above 1000Hz during recording and then boost frequencies below 400Hz during playback. The result is a fairly flat frequency response from 20Hz to 20kHz for expensive recorders and a not so flat frequency response from 50Hz to 9kHz for less expensive recorders.

The final problem in designing a recorder is erasing any signal already on a tape before recording a new one. In most systems this is achieved using a head with a wide gap and either a DC erase signal or the same ultrasonic signal used for the bias.

The principle behind erasing a tape using an AC signal is worth explaining because it is used in bulk erasers and head demagnetisers. If a portion of the tape (or anything else) is kept in an alternating magnetic field its domains are continually being magnetised in opposite directions. If the alternating field is slowly reduced then each domain stops being affected at a different point in the cycle, and hence the final result is an almost random alignment of all the domains on the tape – and a net zero magnetic field. In a tape recorder the slow reduction in the field is achieved by



Integral cassette decks were once a cheap alternative to disk drives. Sharp's early PCs used them long before Amstrad.

moving each portion of tape away from the gap in the erase head. Bulk tape erasers and head demagnetisers work in the same way; the alternating field is usually derived from the 50Hz mains and the reduction in the field is obtained by physically moving the bulk eraser or head demagnetiser away from the object being demagnetised.

Digital recording

The method described above is capable of recording analogue signals such as speech or music, but for computer use we are interested only in recording sequences of bits. To design a tape recorder for computer use from scratch the most obvious method would be to use the two extreme states of magnetisation; ie all the domains with their north poles pointing to the right and all the domains with their north poles pointing to the left, to represent zero and one. (This method of recording is known as 'saturation recording' because the tape is never partially magnetised – it is always fully 'saturated' in one direction of magnetisation or another). Clearly the electronics used to write and read a saturated recording can process the signal in a much more direct way to obtain the best performance. As a result computer tape decks achieve a much higher density of storage and are very reliable. Computer tape decks also move the tape past the head much faster than domestic audio decks to improve the data transfer rate.

Which recorder

Audio tapes for program and data storage are prone to errors. However some people claim to use audio tape regularly without any trouble and others never seem to be able to load a tape. There is no doubt that some computers use recording and reading methods that are inherently reliable – the Spectrum, BBC Micro and Dragon are notable for the robustness of their tape interfaces; one notably bad performer was the Oric which never got to work well at its higher data rate.

It is also true that some cassette recorders work well with one machine and badly with others. Efficiency seems to depend on what sort of recording defect the computer is sensitive to. For example the Dragon seems to be sensitive to level changes in the leader tone and so a tape recorder with automatic volume control tends to upset the Dragon while it works well with, say, the BBC Micro.

If you are looking for a cassette deck to use with your computer then the following points are worth keeping in mind:

1) A cheap tape recorder will often work better with a computer than an expensive one. The reason is that expensive machines tend to have more signal processing circuits and filters and this makes them poor reproducers of the square wave signals generated by most computers. There is also a tendency for complicated filters to introduce frequency dependent phase shifts that are inaudible in music but devastating to a computer trying to measure pulse lengths.

2) A stereo machine is generally unsuitable for computer use. The two channels are recorded as two separate tracks and any phase differences that are caused by misalignment of the heads results in a very poor mono signal if the two channels are combined. The only reliable way of using a stereo tape is to use only one of the channels, ie. either the right or the left.

3) A tone control is a feature that you can do without for computer use – if present always set it at a level that boosts the high frequencies (treble) as much as possible without producing too much background hiss.

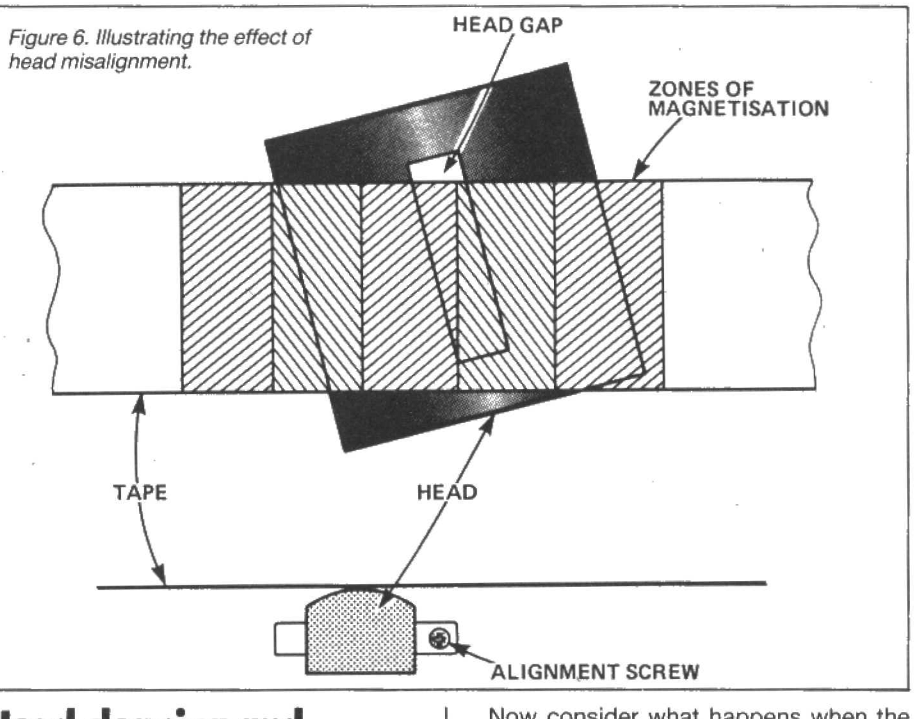
4) Avoid machines with automatic volume control. When recording a computer tape the automatic volume control reacts to boost quiet passages and loud passages and this is not what the computer needs. A manual control can be set to a record and and playback level that results in the most faithful reproduction of the signal tones.

There are other features which make a recorder suitable for computer use, eg a pause control, a tape counter etc. but the only way to discover whether or not a particular recorder will work well with your computer is to try it. The main thing is not to think that the more you pay the more likely it is work: cheap recorders are likely to be suitable for reasons that their designers probably thought of as failings!

out of the head. Instead the head forms a complete magnetic circuit and the magnetic field is completely contained within it. As magnetic tape is coated with magnetic material it is not surprising that heads become dirty. A similar argument applies to the way that the head picks up the magnetic field on playback. The result is that new recordings are made at a much reduced level and old recordings playback at a much reduced level. Early signs of a dirty head are a reduced frequency response that makes the tape sound not only quieter but muffled. The solution to dirty tape heads is to clean them. Tape head cleaners are cheap and reliable and regular cleaning will stop dirt from accumulating and causing trouble. Head cleaners come in two sorts, wet and dry. The wet variety uses alcohol as a solvent and on balance this is the type that I prefer as it seems to be more effective and causes less head wear.

The second most common problem is head misalignment. This manifests itself as difficulty in loading other people's or commercial tapes. To understand why head alignment is important consider a tape recorded on a machine in which the head gap goes at 90 degrees to the edge of the tape. The resulting pattern of magnetisation takes the form of strips also at 90 degrees to the edge of the tape (see **Figure 6**).

Figure 6. Illustrating the effect of head misalignment.



Head cleaning and alignment

If your normally reliable tape recorder suddenly starts to give errors when loading programs and data the first thing to suspect is a dirty head. If you followed the description given earlier or how a tape head works you will appreciate the importance of the head gap in producing a localised magnetic field. Now if the head gap is bridged by some magnetic material then the magnetic field will not be forced

Now consider what happens when the tape is read using a head that is not at 90 degrees to the edge of the tape. The head gap will cross over a number of magnetic strips at the same time and so will be influenced by all of them. The result is as if you were using a head with a larger gap – the frequency response goes down and this can make reading a tape impossible. Of course if a head is out of alignment it will read tapes that it has written perfectly well but not tapes written by other correctly or differently aligned machines.

The way to get rid of misalignment is to

realign the head! Many recorders have a hole drilled in the case that will allow a small screwdriver access to the head adjusting screw, shown in **Figure 6.** By turning this the angle that the head makes with the tape can be altered but beware, if you turn the screw too many times it will come out and you will then have to take the machine apart to put it back. The only difficulty in adjusting the head is knowing when it is correctly aligned. You can buy alignment tapes that contain a very high frequency tone such that the output of the recorder is very sensitive to head alignment, but for computer use this is usually unnecessary. Simply play a commercial music tape, preferably one that contains a lot of high frequencies and adjust the head alignment to produce the clearest sound possible. You can also use a commercial computer tape for the same purpose if it starts with a high frequency leader. Of course if you alter the head alignment you will discover that any tapes you recorded with a different head alignment will be difficult, if not impossible, to read.

If you are having difficulty reading a commercial tape then it is sometimes worth altering the alignment of your machine to produce the clearest tone. In one case I had to adjust the head alignment differently for different sections of a BBC Micro tape to make it load – I have no idea how a mixture of head alignments came about when the tape was made – and this method did load an otherwise unusable tape. Finally, if you do alter the alignment of the head to read a tape remember to put it back as it was.

There are other problems that occur in connection with tape heads. In particular it is possible for a head to become magnetised. Tape head demagnetisers are sold to cure this complaint but in my experience head magnetisation is not a problem with computer use. Head wear is another matter. Tape heads are made of relatively soft metal and they do wear out. It is fairly easy to replace a tape head – they nearly all use the same mounting – but if you chose a cheap tape recorder it probably isn't worth the trouble.

Which cassette?

Most audio tapes, apart from the very cheapest, work well in computer applications. There is no need to buy superior quality tapes – save them for your hi-fi system. But beware of poor quality cassette housing. A bad case can make a tape jam and break and this will lose your data forever. Don't use C90 or C120 tapes for they are too thin to survive much repeated use.

Looking after computer tapes is much the same as looking after audio tapes only it is even more critical. The main problems are heat and magnetic fields – avoid sunny windowsills for storage and be careful where you put the telephone if it has a bell! If you store tapes for a long time then it is worth rewinding them occasionally to avoid 'print through'. When different portions of tape are kept in contact for any length of time the magnetic field of one portion can affect the other. This can sometimes be heard as a 'pre-echo' of an imminent loud passage on music tapes. If you rewind the tape it alters the position of the tape on the spool and so minimises print through.

A bulk tape eraser will erase an entire tape in one operation taking a few seconds. Not only is it quick but it produces a tape that has less background noise than one erased by a machine's erase head and so might improve the reliability of your data storage. I use a bulk eraser regularly – but mainly on video tapes!

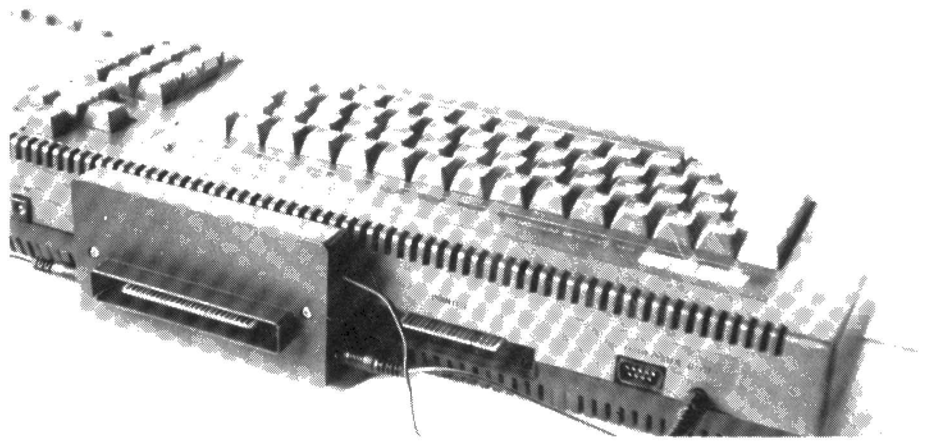
Audio tape and the future

It is difficult to believe that audio cassette recorders can sustain the personal computer's need for fast reliable storage for very much longer. Indeed they are not fast and in many cases they are not even reliable; they can only be seen as a stopgap until something better comes along. A number of better storage devices are already available and it is only a matter of time before audio cassette storage is nothing but a museum exhibit. The future of digital tape recording is another matter and it would takle a brave man to say that tape will be phased out altogether – after all some people still use punched cards!

***Next month** – Disk storage – why, how and what can go wrong.*

A conversation with Amstrad

Speech synthesis is the object of Amstrad's latest addition to their peripheral range. Peter Green phonemes home . . .



Speech synthesis has never been easier. Simply get yourself an SPO256 chip – £14.00 from the Maplin catalogue – add a little address decoding, low-pass filtering and amplification, and your computer can start chatting away. This is what Amstrad have done in their SSA-1 speech synthesiser.

Until recently the only way of producing computer speech was to digitise a real voice, strip away as much of the information content as possible to reduce data storage without losing legibility, and store it in RAM. This meant that only fixed phrases could be used, and it's a very inflexible, almost useless procedure. (Of course, Ghostbusters and Impossible Mission on the Commodore 64 show what *can* be done using this method.)

In contrast, the SPO256 makes it possible to say almost anything in the English language. Instead of storing digitised phrases, the chip contains a digital filter that models the human vocal tract. An on-chip ROM can excite this filter to produce allophones, the discreet sounds that make up human speech. And any word can be spoken by stringing the right allophones together. There are 59 allophones, and by including five pauses of different lengths we get 64 possible outputs.

So speech synthesis involves sending a stream of one-byte numbers to the chip from the host computer, using a simple handshaking system. The SSA-1 occupies address &FBEE in the Amstrad I/O map. How annoying for software houses that the rival product from DK'Tronics, announced a few months back, uses the same chip but puts it at &FBFE.

Hard facts

The SSA-1 hardware is supplied in a colour-matched case very similar to that of the DD1 disk drive interface. It even has the same ineffectual ribs that are supposed to hug the contours of the computer's casing, but don't.

The SSA-1 will also function as an external amplifier for the stereo sound channels of the 464/664. A flying lead plugs into the sound jack at the rear left of the computer to pick up the signal. There is a master volume control, and two speakers plug into either side of the interface. Helpfully, the manual explains that the left speaker plugs into the left hand side . . .

The interface is provided with a through connector so that the disk drive, and any future peripherals, can also be fitted (although with the Arnor ROM board fitted too, the power lead on my machine is heading for full stretch!).

Soft touches

Machine code programmers need to know little more than the port address and the simple details of handshaking to drive the unit, but for BASIC programmers, and those requiring text-to-speech facilities, a det of RSXs (Resident System Extensions, or ¦ commands) is supplied on tape. Amstrad says it was done to minimise the cost, but the Amstrad concept (according to their literature, at least) was that peripherals could store their external commands in an on-board ROM and reserve any RAM workspace on power-up (like the disk drive).

There are nine new commands provided, and they can be used either from BASIC or machine code (both techniques are explained fully in the usual excellent Amstrad domumentation). The simplest is ¦SAY, followed by a string variable address. This converts the string into a series of allophone numbers and feeds them to the sound chip.

An alternative mode is accessed using ¦ECHO plus a parameter from 0 to 4. Echo mode is disabled by 0. In Mode 1, all print output enclosed by the ¶ mark (reverse quote) is spoken and displayed. (The ¶ must be inside the normal "" marks.)

Mode 2 speaks all screen output, including listings, "Ready" and error messages, but does not display it. Mode 3 speaks *and* displays everything, while Mode 4 speaks but does not display text output between ¶ marks.

The manual mentions that ¶ is ASCII 60H, but not what key it is on. It's shifted-backslash, ie shift plus the key immediately above CTRL.

The text-to-speech is implemented quite intelligently. For example, numbers shorter than 10 digits are interpreted to their English equivalents – "two thousand and ten" rather than "two zero one zero". Conversion is done by a set of algorithms that cope with most vagaries of English pronunciation, but which can come unstuck. For example, "goodbye" comes out as "good-dbee". In cases like this you can try alternative spellings to fool the software: "goodbiii" works better here.

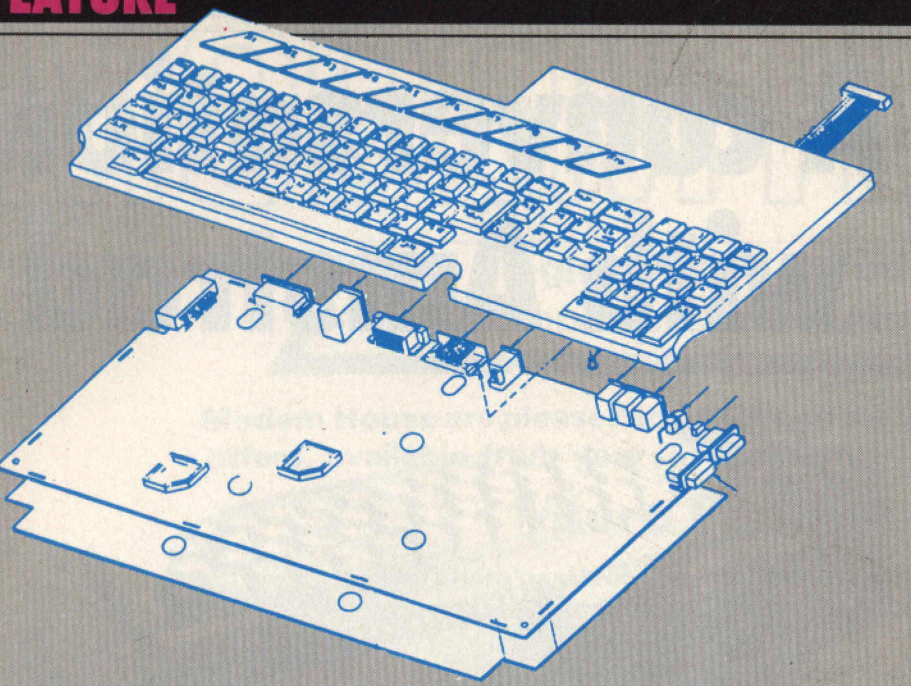
If all else fails, you can get the pronunciation right by shifting down a level in the system and sending allophone numbers directly to the buffer using APHONE. The allophone list for the title of this article is 11,32,2,25,23,58,2,13,25,45,42,12,12, 44. ¦ROOM can be used to find out how many free locations are left in the allophone buffer; whether the chip is currently sounding an allophone; and whether the speech interrupts are active. QUIET flushes out the allophone buffer and silences the speech chip.

Shifting down yet another level in the system, we come to ¦SPON and ¦SPOFF. The allophones are transferred from the buffer to the hardware under an interrupt-driven event routine.

All the commands are explained fully in the manual, with copious examples including a BASIC speaking clock.

A speech system as simple as this cannot hope to do better than produce a Dalek-type voice. Nevertheless it often gives a perfectly eloquent output, especially when the text-to-speech is bypassed and words are tweaked using APHONE.

Things like the spoken listings are almost unintelligible because of the equal emphasis placed on each syllable and the constant rhythm, with none of the ebb and flow of human speech. Nevertheless, I'm sure we'll be seeing games with speech options quite soon.



The diagrams on the left and right are taken from genuine Atari production plans of the new 520ST. John Lambert has studied the full schematic and says the ST will set new standards in performance and computer design.

The Atari blueprints

Authenticated production plans of Atari's 520ST computer – including a full circuit diagram – have been passed to *Electronics and Computing*. The plans show that the ST is a superbly designed machine and that Atari is now ready to go into production at its Far East manufacturing facility.

The finished machine is *not* the same as that shown recently at the Hanover show and described in other magazines. The GEM operating environment will be on disk, but the production machine has no TV output – Atari will supply it with a monitor. It also has only 16K internal ROM.

The plans show that Atari has used an original and elegant method of memory management which should make the ST faster than any other PC on the market – in any price bracket.

The ST uses very few chips. The logic is packed into a few large scale custom chips, one of which – code named GLUE – presides over the ROM as well as ST peripheral devices. GLUE is at the centre of the ST design.

And the hard disk interface is a fast parallel port which should be able to transfer data at speeds in excess of 1Mbyte per second.

There will be turmoil in an already troubled computer industry when the ST is launched in the 'States in June. The specification outstrips nearly all computers up to, and including, the IBM PC.

The only UK machine in this bracket with a 68000 series processor, the Sinclair QL, is in for a rough ride.

For £699.99 a buyer will get a very fast 16/32-bit computer (Sir Clive's QL uses a cut down 8/32-bit chip which has 30% slower throughput); 512Kbytes of RAM; a 3.5″ floppy disk drive plus a hard disk interface; mouse driven GEM operating software; DR Logo and BASIC; a very high definition colour display; three channel sound; parallel and RS232C ports; and a Midi synthesiser interface. Specifications always look better on paper, but if the ST lives up to its promise it will be the final nail in the coffin of the 8-bit 64K micro, and may do everything that Sinclair set out – but has as yet failed – to achieve with the QL.

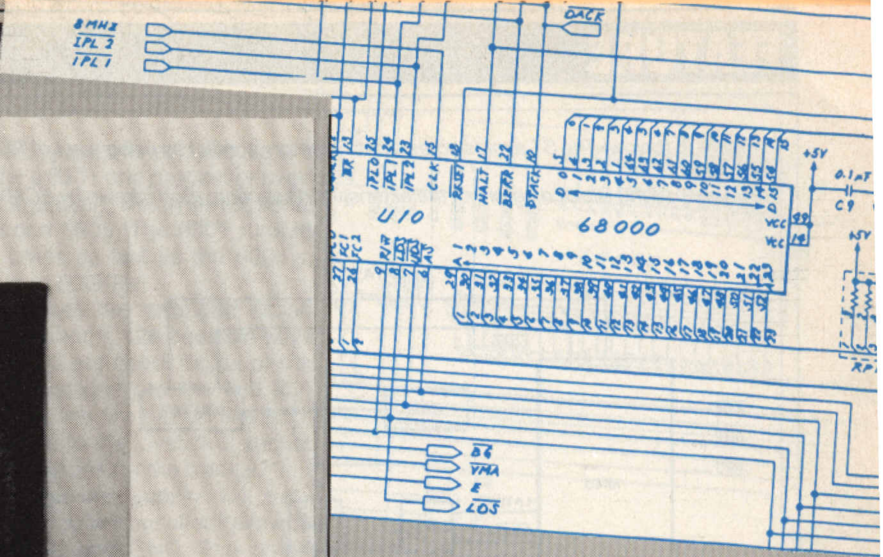
Design of the ST range, 130ST (128K), 260ST (256K) and 520ST was started over a year ago, long before Jack Tramiel took over the running of the company, but his hand is very much on the final product. His often quoted commandment "It will be done" has meant that at least the first batch of machines will have GEM on disk, rather than in ROM, in order to get them to the shops in time.

BASIC and Logo will also be held externally. These languages are being supplied by Digital Research, writers of GEM, the BASIC being a version of that originally supplied by Metacomco for the IBM PC back in 1983 – that's hardly state-of-the-art.

In the last year chip prices have dropped dramatically and this has affected the design. A year ago RAM, particularly the 256K chips, were very expensive compared with the 64K chips used in the 130ST and 260ST. These are now down to around £2.50 in the Far East, where the ST was designed and will be built. This means that the 520ST does not cost much more than the 130ST (16 memory chips), and less than the 260ST (32 chips). For this reason the 260ST is unlikely to appear.

This price drop also means that ROM chips are now very expensive. the 520ST has 16K of ROM internally, in two 64K by 1-bit chips. The cost of the original specification of 192K of ROM in six 32K by 8-bit chips, is now greater than the extra memory and the disk drive put together. This puts a question mark over the future of the 130ST with its comparatively small memory.

The only other omission on the production models is a means of driving a domestic TV. This is another result of the haste to get it out on time and it may be included on later versions.



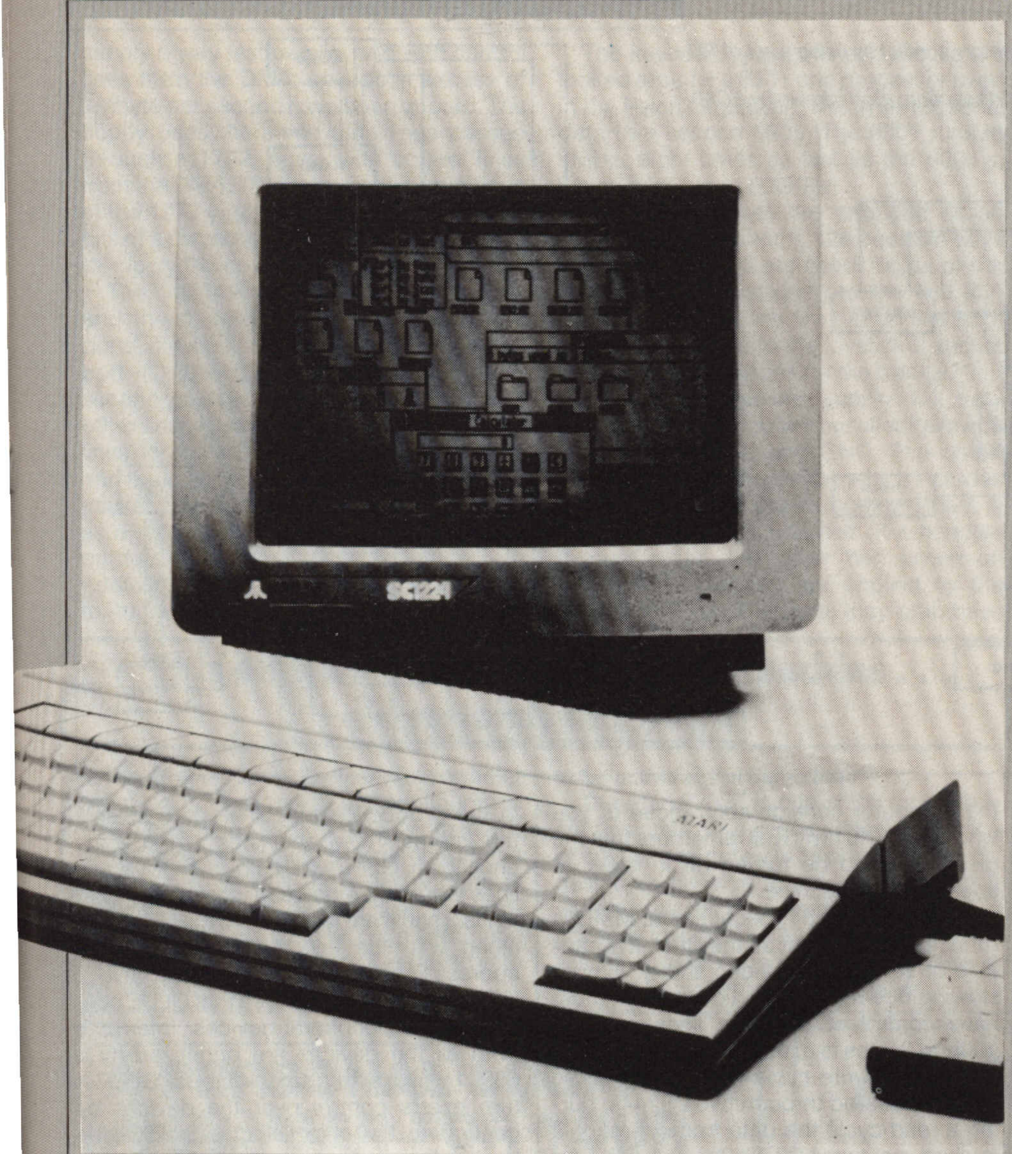
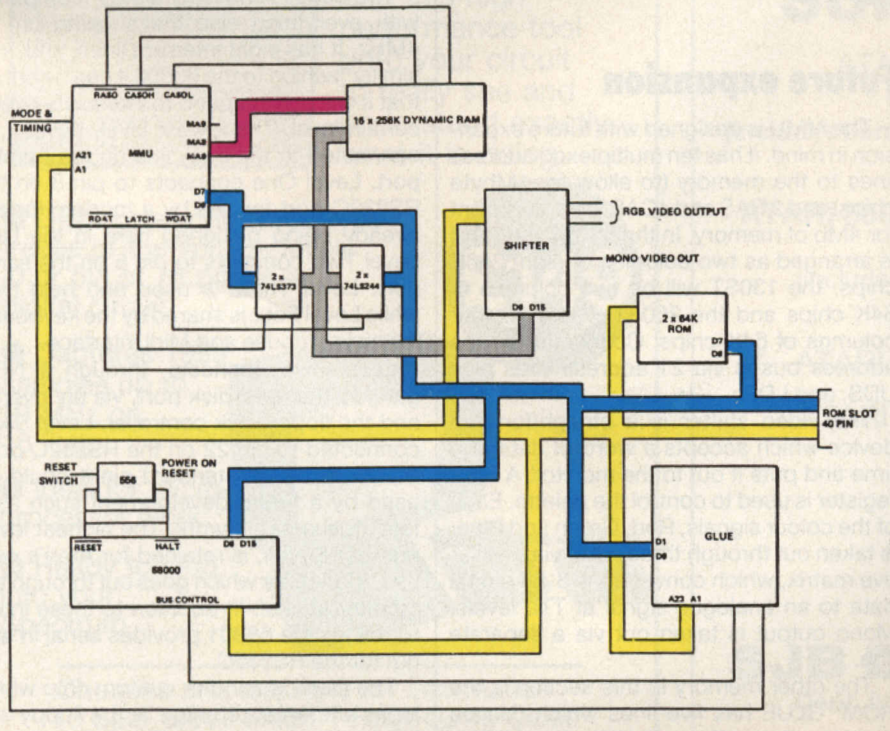


Figure 1. Block diagram of the 520ST excluding I/O section (drawn by ECM).



GEM of a display

The display has three modes: 320 x 200, sixteen colour; 640 x 200, four colour (from a palette of 512 colours); and 640 x 400 two colour – all of which would be difficult to display on a TV in any case. Atari is therefore bringing out a range of low cost monitors to match the ST.

At the launch of the QL Nigel Serle held up a list showing what it would cost to have a system with the same specification as the QL. To bring the QL up to the standard of the 520ST, if it could be done, would cost in the region of £1350. Enough of a price difference to buy a few programs. Trying to do the same thing with a system based on the new BBC would be even more expensive.

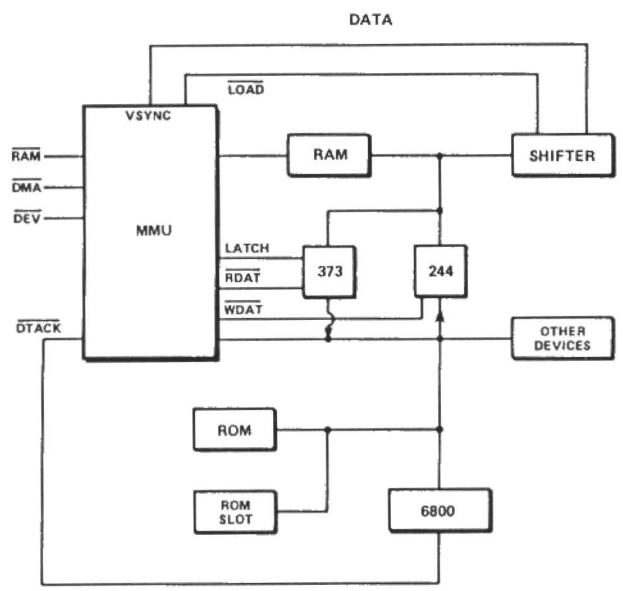
"The ST could be the final nail in the coffin of the 8-bit 64K micro"

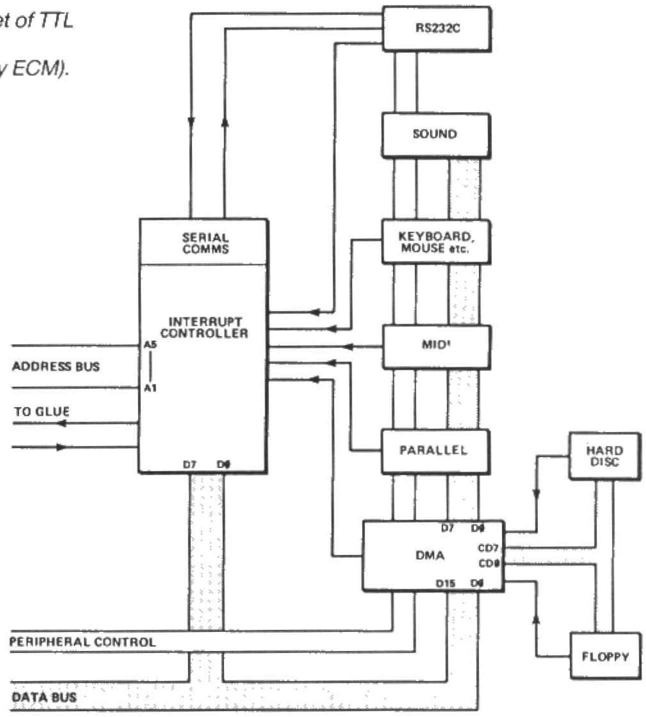
Both of these machines are aimed at the home/small business market and in this area GEM is going to be a good selling point. There are a lot of people who know very little about computers other than that they think they ought to have one. Icons, pull down menus and mice (mouses!) are the buzz words of the moment and they sell computers.

From the outside the 520ST is a very impressive machine. As well as the standard QWERTY keyboard, above which are ten function keys, there is a cursor pad, with additional keys Help, Undo, Insert and Clear Home for use with GEM. To the right of this is a full numeric pad with its own Enter key and mathematical functions. Four different versions of the keyboard will be produced: USA, UK, French and German.

Viewed from the back this machine is no less impressive. On the right hand side is a 40-pin ROM socket. Along the back, from the right, are a 19-pin plug for a hard disk; a 14-pin DIN socket for the floppy disk; a 25-pin D type plug for RS232C; a 25-pin D type socket parallel port; a 13-pin DIN socket for video and audio in and out; two

Figure 2. (below) The ST avoids screen/data memory conflict by using a set of TTL latches.
Figure 3. (right) Block diagram of the extensive I/O section. (Both drawn by ECM).



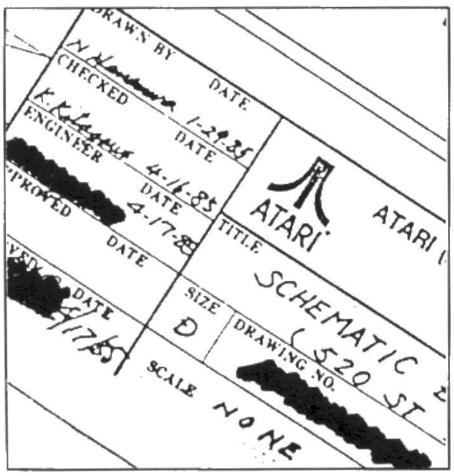


5-pin DIN sockets for Midi in and out; a 7-pin DIN power plug (to the external supply); an on/off switch; and a reset switch. On the left hand side are two 9-pin D type sockets for joysticks. (Joystick 0 also takes the mouse). Enough ports to take your breath away.

The real eye opener is inside the machine. Its workings can be grouped into two areas: CPU-memory-video, and peripherals, all presided over by a custom chip code named GLUE (presumably because everything is stuck through it). This is illustrated in **Figure 1.**

The CPU is a standard Motorola 68000 running at 8MHz. The MMU (Memory Management Unit) is a custom designed chip, running at 16MHz and the output to the video is looked after by another custom chip, a video shifter running at 32MHz.

A common problem in all computer design is that an area of memory has to be shared between the CPU and the video. A number of solutions are used: in the QL (and the Spectrum) the CPU is denied access to the screen memory when the video is being put out which slows down



the computer considerably; those of you who had a ZX81 will well remember that the display had to be turned off in order to speed up the machine.

But on the 520ST, because the MMU is running twice as fast as the CPU, it is capable of servicing the video shifter without slowing down the CPU. To do this it must have control of the data lines and the method it uses is extremely simple:

During a write cycle the data from the CPU is buffered through two LS244s; during a read cycle the data from memory is latched via two LS373s for the CPU (or other devices) to pick up when it is ready. The data bus is now free for the video. (see **Figure 2**).

Future expansion

The MMU is designed with future expansion in mind. It has ten multiplexed address lines to the memory (to allow for 1Mbyte chips) and 2 RAS and 4CAS lines, sufficient for 4Mb of memory. In the 520ST the RAM is arranged as two columns of eight 256K chips, the 130ST will be two columns of 64K chips and the 260ST would be four columns of 64K chips. Connection to the address bus is via 21 address lines plus UDS, and LDS.

The video shifter is a straightforward device which accepts a word of data at a time and puts it out to the monitor. A 5-bit register is used to control the palette. Each of the colour signals, Red, Green and Blue, is taken out through three lines, via a resistive matrix, which converts the 3-bit digital data to an analogue signal at TTL levels. Mono output is taken out via a separate line.

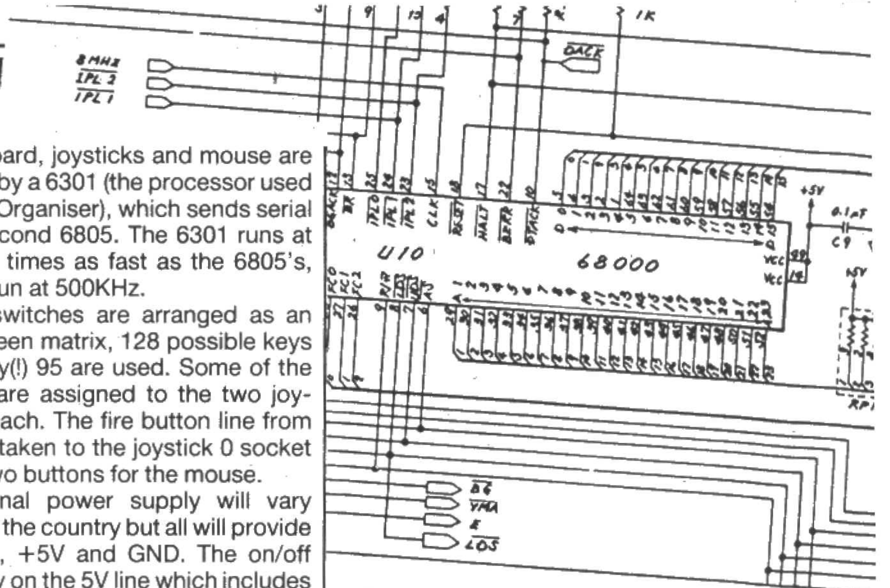
The other memory in this section is the ROM. GLUE has five lines which can be used to select different 64K banks of ROM. In the production machine line 0 selects the internal ROM and lines 1 and 2 are unused. If GEM was held in ROM these lines would then be used, providing 192K. Lines 3 and 4 are taken out to the ROM slot, along with the full 16-bit data bus, 15 address lines, UDS, LDS, +5V and GND.

The other important area that GLUE controls is the peripheral devices. Interrupts from all the devices are prioritized by a 68901 which in turn signals GLUE. GLUE in turn exerts control over the devices by generating the chip select signal. Of the three interrupt lines on the CPU only two are used with IPLO held high, and IPL1 and IPL2 taken directly to GLUE **(Figure 3).**

The 68901 runs fairly slowly (compared with everything else that's going on) at 4MHz. It has eight interrupt lines, and, in a similar fashion to the 68000 it can be set so that it will only respond to interrupts over a certain level. The lowest level, Level 0, is connected to the busy line on the parallel port. Level One connects to pin 8 on the RS232C port for use by a modem (one is already being designed here in the UK). Level Two connects to pin 5 on the same port. Level Three is used and held high while Level Four is shared by the keyboard, joysticks, mouse and Midi interface.

Level Five connects, through a NOR gate, to the hard disk port, via an inverter and the floppy disk controller. Level Six is connected to pin 22 on the RS232C port. As this is a non-standard pin it could be used by a future development such as a fast local area network. The highest level, number Seven, is retained for Atari's own use, the line for which goes out through the monitor socket. In addition to these interrupt lines the 68901 provides serial in and out for the RS232C.

The DMA is another custom chip which looks after mass storage, ie the floppy and hard disk interfaces. The hard disk is



simply a very fast parallel port with control lines from the DMA; speeds in excess of 1Mbyte per second have been quoted. The floppy uses the same data lines passed through a 1772, again with control lines from the DMA. To overcome the limitations of the 1772 (a general purpose chip) three bits of a parallel port on a YM2149 are used to supply the drive and side-select lines.

The YM2149 has two 8-bit parallel ports and three analogue outputs. One of the ports directly drives the external parallel port (Centronics) with one line of the second port providing strobe. Of the four other lines on the YM2149 two are control lines for the RS232C port, one is unused and the other goes out through the monitor socket as a general purpose output, used as a software switch. The three analogue outputs are simply joined together, with Audio In, and passed via a single transistor to Audio Out.

The keyboard, joysticks and mouse are looked after by a 6301 (the processor used in the Psion Organiser), which sends serial data to a second 6805. The 6301 runs at 4MHz, eight times as fast as the 6805's, which both run at 500KHz.

The key switches are arranged as an eight by sixteen matrix, 128 possible keys of which only(!) 95 are used. Some of the spare keys are assigned to the two joysticks, five each. The fire button line from joystick 1 is taken to the joystick 0 socket to provide two buttons for the mouse.

The external power supply will vary according to the country but all will provide +12V, −12V, +5V and GND. The on/off switch is only on the 5V line which includes a line filter inside the computer. Reset is handled by a 556; one half of it handles manual reset, the other half power-on reset. Both Outputs are applied to the RESET and HALT lines as they both have to be active at the same time.

"It is difficult to find fault with this machine . . . I'd certainly buy one".

The Midi ports are basically just another serial port. They use a pin configuration which has been standardised by the music industry for connection of a computer to a synthesiser, and all the work is done in software. Midi In passes via an opto-isolator to the RX Data pin of a 6805, and through to Midi Out, where it is mixed with the signal from TX Data.

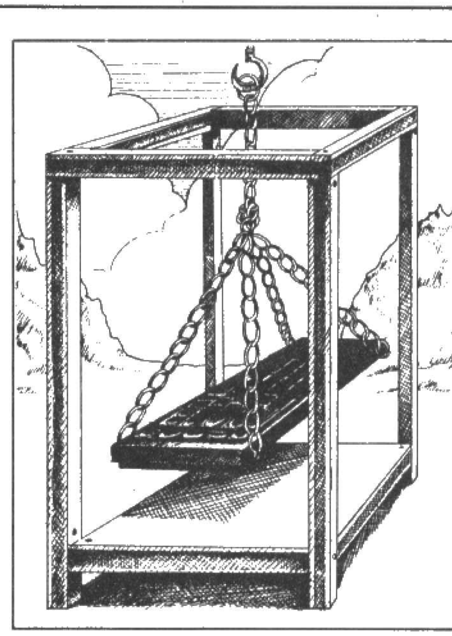
The Atari 520ST is one of the most elegant designs I have seen and far in advance of anything remotely in its price bracket. Its designers deserve a bonus. The effect it will have on the home computer market is not easy to judge but by Christmas, when it will have established a software base, it looks to be unbeatable at the price.

In conclusion

It is difficult to find fault with this machine, though a few more months development would have made a difference. The only extra facility which perhaps should have been added is some form of bus expansion. As the machine stands it can handle a lot of extra memory but there is no way it could be added without opening the computer and getting the solding iron out.

The 64K dollar question is would I (a poor underpaid journalist – Editor take note) go out and spend money for one? To which the only answer is "Try and stop me!"

QSHELL



Part 2 of Adam Denning's MS.DOS lookalike front end for QDOS.

Last month we looked at the SuperBASIC extension procedure which loaded and executed the shell program; now we must write QSHELL itself. Unfortunately, there is a great deal of source code (70K at the moment – and it's not even finished), so it will be spread out over more than one issue. Only at the end of the series can it be usefully assembled and run, as numerous routines are borrowed by others, variables are often shared, and so on. Here we look at the initialisation routines and the start of the command line interpreter itself.

You'll remember from last month's issue that QSHELL is activated by a BASIC procedure which also puts some data on the job's stack. This data comprises three channel IDs and the address of the KEYTAB pointer table. To tell QSHELL that it has four items of data on its stack, a word of four is placed at the very top.

Before it does anything else, then, QSHELL must check that it has been activated in this way. It does so by pulling a word off the stack and subtracting four. If this results in zero, everything is okay, otherwise it must kill itself with an error. We've chosen 'bad parameter' as an appropriate error, so we execute FRJOB with this code in D3.

If all is well, the system should be asked for some memory. We get this from the common heap, by calling ALCHP. The amount of memory required is enough to store three command lines (each of which cannot be more than 152 bytes long) and the default device name, which can be near enough 40 bytes long. If this request succeeds, we then set about storing various values in the static data area. Using the post-increment addressing mode, this is particularly easy. All four stacked items are stored, followed by the address returned by ALCHP, which is the address of the first byte available in the heap allocation.

The heap area is now split up into four different sections – the default device name and three command line stores – and the address of each is saved in the statics. By this time A1 will have been incremented enough to point to 80 spare bytes, the first of which has the label EXCP_TAB. This will be the trap redirection table, so the next 20 long words must be filled with the address of our exception handling routine, FOUL_UP (it was called something else). We then call MT_TRAPV to set this table for QSHELL, and do the same again with 0 in D1 to set it for the SuperBASIC interpreter as well. As a sort of extra security, the priority of the BASIC interpreter is set to 0 with MT_PRIOR.

With the manager section complete, we can concentrate on the aesthetics. Routines WINDOW0 to WINDOW2 are called to set up the colours and sizes of the three screen windows and then we print a prompt to channel 0. Before entering the command loop P_DEFAULT is called (this sets the default directory device) followed by GETCLOCK to load and start a click – if we can find it.

The window routines are all very simple. The address of the respective parameter block is loaded into A1 with LEA, and a branch is made to SET_WINDW. This uses the first two bytes in the parameter block to define the border, and calls SD_WDEF to set this and move the window to its designated location. SD_SETPA, SD_SETST and SD_SETIN are then called in turn to set the paper, strip and ink colours, and finally the window is cleared by calling SD_CLEAR. Makes life rather easy, these QDOS trap routines.

P_DEFAULT is the routine which sets the initial default device. It starts up by trying to open the directory on a device called 'FLP1_', which is the device name of a CST floppy disk interface. If this succeeds, it then stores this name in the common heap area allocated. Otherwise, it uses microdrive 1 instead, and stores 'MDV1_'. Obviously, if you have a disk interface from another manufacturer, you should replace 'FLP1_' with whatever is required ('FDV1_', 'FDK1_', etc).

Notice how the default device name is moved into the common heap area with the help of a utility routine called STRING_MV. This uses a further routine, APPEND_S, to append a null string (ie nothing) to the device name, thus effecting the move. At the moment, the various data definitions and utility routines are scattered around the source file, which is considered bad practice, so try to move all strings, messages and so on down to the bottom of the file when you finally assemble it.

GETCLOCK is a routine whose sole purpose it to load and execute a program called SCLOCK. There is another routine elsewhere in the program which also loads and executes jobs, and which should really incorporate GETCLOCK. But unfortunately, the fact that SCLOCK must be activated with a priority of 1 *and* create no error if the file isn't present caused me to initially write it as a separate routine, and I haven't got around to changing it yet.

GETCLOCK works by appending the default device name to the file name and then opening this file for input. If this fails, the clock program isn't present, so we must leave GETCLOCK straightaway. If it succeeds, the file's header is loaded into memory and its length and default data space size is read. Using this information, we can then ask MT_CJOB to create a job for us. If this causes an error we leave, or otherwise load the file into the job area thus created by QDOS using FS_LOAD, close the file using IO_CLOSE, and then activate the job with a priority of 1 and a timeout of 0 with MT_ACTIV.

OPEN_DEF simply appends the filename pointed to by A3 to the default device, giving us a valid filename. It then tries to open this file for shared input using IO_OPEN, and returns with A0 holding the channel ID if it succeeded, or D0 holding an error code and the zero flag reset if it failed.

STRING_MV is a general purpose utility routine to move a string pointed to by A2 into an area pointed to by A4. It does this by loading the address of a null string (zero length string) into A3 and then falling through to APPEND_S, which joins together the strings pointed to by A2 and A3 at the space pointed to by A4. APPEND_S is straightforward enough, but notice how it takes special provision to

ensure it works correctly if either the string at A2 or A3 (or both) is null.

That's all the initialisation there is, as the next few routines form the start of the command line interpreter. We'll explain these in detail next month, but briefly, COMMAND gets a line of input from channel 0, converts all groups of spaces into single spaces using REMOVE, and then stores the command line in one of the command lines stores requisitioned from the common heap. It then passes this line to SPLIT_CLN, to be split into its component parts, and then to DO_COMMAND which recognises the command and calls the appropriate command routine, or prints an error message if it is not recognised.

GET_COMLN is the subroutine which does the actual collection of the command line from the operator, and allows you to re-edit it if it is returned from other routines.

Finally this month, we have the exception processing routine, FOUL_UP, which is called if an unexpected exception (such as pressing CTRL-ALT-7) occurs. The version here is rather simplistic, as it merely prints out a message to the console and returns, but it would make more sense to take the time to expand it into a more useful recovery routine which re-invoked the shell, or whatever.

Next month, we'll write the command line interpreter and command routine table, which decides if a command makes sense or not, and what to do with it if it does.

LISTING 1. QSHELL initialisation routines.

```
* The SHELL program...a QDOS front end
* By Adam Denning (C) 1985 Adam Denning


linefeed  EQU     10
escape    EQU     27
hdrlen    EQU     64                  directory header entry length
sector    EQU     512                 length of a file system sector
deflen    EQU     40                  maximum device name !ength
maxline   EQU     150                 maximum command line length


          GET     'adv1_header_asm'
          SIZE    700           ??

* This job is loaded and executed from the SuperBasic interpreter, with a timeout
* of -1 and a priority of 16 (at the moment). It will find some info on its stack
* waiting for it :    Channel 0 ID  --  long word
*                     Channel 1 ID  --  long word
*                     Channel 2 ID  --  long word
*                     Address of KEYTAB --  long word
*                     Binary 4  -- word
* A7 points here -------->> (well, to the 4 actually)


          BRA.S     START_S
          DC.L      0
          DC.W      $4AFB
          DC.W      6
          DC.B      'SHELL '

START_S   MOVE.W    (A7)+,D1
          SUBQ.W    #4,D1
          BEQ.S     OK_SHELL
          MOVEQ     #ERR_BP,D3
ABORT     MOVEQ     #-1,D1
          QDOS      MT_FRJOB,1

OK_SHELL  MOVEQ     #-1,D2
          MOVE.L    #512,D1
*
          QDOS      MT_ALCHP,1
          TST.L     D0
          BEQ.S     BEGIN_IT
          MOVE.L    D0,D3
          BRA.S     ABORT
BEGIN_IT  LEA.L     STATICS,A1
          MOVE.L    (A7)+,(A1)+
          MOVE.L    (A7)+,(A1)+
          MOVE.L    (A7)+,(A1)+
          MOVE.L    (A7)+,(A1)+
          MOVE.L    A0,(A1)+
          CLR.L     (A1)+
          ADDA.L    #deflen+10,A0
          MOVE.L    A0,(A1)+
```

```
CLR.W     (A0)
ADDA.L    #maxline+2,A0
MOVE.L    A0,(A1)+
          CLR.W     (A0)
          ADDA.L    #maxline+2,A0
          MOVE.L    A0,(A1)+
          CLR.W     (A0)
          LEA.L     FOUL_UP,A0
          MOVEA.L   A1,A2
          MOVEQ     #19,D0
EXCP_MV   MOVE.L    A0,(A1)+
          DBRA      D0,EXCP_MV
          MOVEA.L   A2,A1
          MOVEQ     #-1,D1
          QDOS      MT_TRAPV,1
          MOVEA.L   A2,A1
          MOVEQ     #0,D1
          QDOS      MT_TRAPV,1
          MOVEQ     #0,D2
          QDOS      MT_PRIOR,1
          BSR.S     WINDOW0
          BSR       WINDOW1
          BSR       WINDOW2
          LEA.L     SIGNON,A1
          MOVE.L    CHAN_0,A0
          VECTOR    UT_MTEXT,2,CALL
          BSR       P_DEFAULT
          BSR       GETCLOCK
C_LOOP    BSR       COMMAND
          BRA.S     C_LOOP
```

```
CS_EXIT   MOVEQ     #0,D1
          MOVEQ     #32,D2
          QDOS      MT_PRIOR,1
          MOVEQ     #-1,D1
          MOVEQ     #0,D3
          QDOS      MT_FRJOB,1

SIGNON    DC.W      54
          DC.B      'QSHELL Command

Interpreter V1.0 (C) 1985 Adam Denning',10

WINDOW0   MOVE.L    CHAN_0,A0
          LEA.L     PBLOCK0,A1
          BRA.S     SET_WINDW

WINDOW1   MOVE.L    CHAN_1,A0
          LEA.L     PBLOCK1,A1
          BRA.S     SET_WINDW

WINDOW2   MOVE.L    CHAN_2,A0
          LEA.L     PBLOCK2,A1
```

***A routine to set up a window using a parameter block pointed to by A1. A0 holds the window's channel ID on entry.**

```
SET_WINDW MOVE.B    (A1)+,D1
          MOVE.B    (A1)+,D2
          MOVE.L    A1,-(A7)
          MOVEQ     #-1,D3
          QDOS      SD_WDEF,3
          MOVEA.L   (A7)+,A2
          MOVE.B    8(A2),D1
          QDOS      SD_SETPA,3
          QDOS      SD_SETST,3
          MOVE.B    9(A2),D1
          QDOS      SD_SETIN,3
          QDOS      SD_CLEAR,3
          RTS
PBLOCK0   DC.B      0
          DC.B      0
          DC.W      512
          DC.W      30
          DC.W      0
          DC.W      0
          DC.B      18
          DC.B      7

PBLOCK1   DC.B      0
          DC.B      0
          DC.W      512
          DC.W      214
          DC.W      0
          DC.W      30
          DC.B      0
```

```
         DC.B    4
PBLOCK2  DC.B    2
         DC.B    1
         DC.W    364
         DC.W    12
         DC.W    0
         DC.W    244
         DC.B    4
         DC.B    0
P_DEFAULT LEA.L  DEF_DR1,A0
         MOVEQ   #-1,D1
         MOVEQ   #OPEN_DIR,D3
         QDOS    IO_OPEN,2
         TST.L   D0
         BNE.S   NO_FLOPPY
         QDOS    IO_CLOSE,2
         LEA.L   DEF_DR1,A2
PUT_DEFT MOVEA.L DEFAULT,A4
         MOVE.L  A4,-(A7)
         BSR     STRING_MV
         MOVE.L  CHAN_2,A0
         MOVEQ   #-1,D3
         QDOS    SD_CLEAR,3
         LEA.L   USE_MES,A1
         VECTOR  UT_MTEXT,2,CALL
         MOVE.L  (A7)+,A1
         VECTOR  UT_MTEXT,2,JUMP
NO_FLOPPY LEA.L  DEF_DR2,A2
         BRA.S   PUT_DEFT

DEF_DR1  DC.W    5
         DC.B    'FLP1_',0

DEF_DR2  DC.W    5
         DC.B    'MDV1_',0
NULL_STRG DC.W   0
USE_MES  DC.W    6
         DC.B    'Using

GETCLOCK LEA.L   CLOCK_FL,A3
         LEA.L   D_SPACE,A4
         BSR.S   OPEN_DEF
         BNE.S   NO_CJOB
         MOVEQ   #16,D2
         MOVEQ   #-1,D3
         LEA.L   D_SPACE,A1
         MOVEA.L A1,A2
         QDOS    FS_HEADR,3
         MOVE.L  A0,-(A7)
         MOVE.L  (A2),D2
         MOVE.L  6(A2),D3
         MOVEQ   #-1,D1
         SUBA.L  A1,A1
         QDOS    MT_CJOB,1
         TST.L   D0
         BEQ.S   CJOB_OK
         MOVE.L  (A7)+,A0
         QDOS    IO_CLOSE,2
         BRA.S   NO_CJOB
CJOB_OK  MOVE.L  A0,A1
         MOVE.L  (A7)+,A0
         MOVE.L  D1,-(A7)
         MOVEQ   #-1,D3
         QDOS    FS_LOAD,3
         QDOS    IO_CLOSE,2
         MOVE.L  (A7)+,D1
         MOVEQ   #1,D2
         MOVEQ   #0,D3
         QDOS    MT_ACTIV,1

NO_CJOB  RTS
CLOCK_FL DC.W    6
         DC.B    'SCLOCK'

*Joins string in A3 to default device, using area at
(A4) as workspace. Opens that file for shared input
and returns with D0 and flags holding error code.

OPEN_DEF MOVEA.L DEFAULT,A2
         BSR.S   APPEND_S
         MOVEA.L A2,A0
         MOVEQ   #-1,D1
         MOVEQ   #OPEN_INS,D3
         QDOS    IO_OPEN,2
         TST.L   D0
         RTS

*Moves a string pointed to by A2 to an area pointed
to by A4. Returns with A2 pointing to new area,
A4 pointing to end of new string. A3 and D2 are
smashed.

STRING_MV LEA.L  NULL_STRG,A3

*Joins two strings together
*First string pointed to by A2, second by A3
*Area to put result pointed to by A4, returned in A2

APPEND_S MOVE.L  A4,-(A7)
         ADDQ.L  #2,A4
         MOVE.W  (A2)+,D2
         MOVE.W  D2,-(A7)
         BEQ.S   FS_SD_INT
         SUBQ.W  #1,D2
FS_LOOP  MOVE.B  (A2)+,(A4)+
         DBRA    D2,FS_LOOP
FS_SD_INT MOVE.W (A3)+,D2
         MOVE.W  D2,-(A7)
         BEQ.S   SD_FS_INT
         SUBQ.W  #1,D2
SD_LOOP  MOVE.B  (A3)+,(A4)+
         DBRA    D2,SD_LOOP
SD_FS_INT MOVE.W (A7)+,D2
         ADD.W   (A7)+,D2
         MOVE.L  (A7)+,A2
         MOVE.W  D2,(A2)
         RTS
COMMAND  MOVE.L  CHAN_0,A0
         MOVEQ   #-1,D3
         QDOS    SD_CURE,3
         BSR.S   GET_COMLN
STORE_ED MOVEQ   #' ',D3
         MOVE.L  D3,D4
         MOVE.L  D3,D5
         BSR     REMOVE
         CMPI.B  #1,D1
         BNE.S   HERE_LK
         CMPI.B  #' ',2(A1)
         BEQ.S   COMMAND
HERE_LK  MOVEA.L A1,A2
         MOVEQ   #0,D0
         MOVE.W  (A2)+,D0
         SUBQ.L  #1,D0
         CMPI.B  #' ',0(A2,D0.L)
         BNE.S   NO_TRAIL
         SUBQ.W  #1,D1
         MOVE.W  D1,(A1)
NO_TRAIL MOVEQ   #0,D4
         MOVE.W  COM_LEVL,D4
         ADDQ.L  #1,D4
         CMPI.B  #4,D4

         BNE.S   NOT_OVER
         MOVEQ   #1,D4
NOT_OVER LEA.L   COM_LEVL,A2
         MOVE.W  D4,(A2)+
         SUBQ.W  #1,D4
         LSL.L   #2,D4
         MOVEA.L 0(A2,D4.L),A4
         MOVEA.L A1,A2
         BSR     STRING_MV
REPT_CMD BSR.S   SPLIT_CLN
         JMP     DO_COMND

*Routine to get a command (channel 0 ID in A0,
infinite timeout in D3)

GET_COMLN MOVEQ  #'>',D1
         QDOS    IO_SBYTE,3
         LEA.L   D_SPACE+2,A1
         MOVE.W  #maxline,D2
         QDOS    IO_FLINE,3
COMLN_GET CMPI.W #1,D1
         BEQ.S   GET_COMLN
         TST.L   D0
         BEQ.S   COMLN_OK
         ANDI.L  #$FFFF,D1
         SUBQ.W  #1,D1
         SUBQ.L  #1,A1
         MOVEM.L D1/A1,-(A7)
         LEA.L   L2L_MES,A1
         VECTOR  UT_MTEXT,2,CALL
RE_EDIT  MOVEQ   #'>',D1
         MOVEQ   #-1,D3
         QDOS    IO_SBYTE,3
         MOVEM.L (A7)+,D1/A1
         MOVE.W  #maxline,D2
         QDOS    IO_EDLIN,3
         BRA.S   COMLN_GET
COMLN_OK SUBQ.W  #1,D1
         LEA.L   D_SPACE,A1
         MOVE.W  D1,(A1)
         RTS
L2L_MES  DC.W    30
         DC.B    10,'Line too long
           - please edit:',10

*The rest of the code - and there's a lot of it - goes
in here
*Followed by exception processing routing:

FOUL_UP  MOVEM.L D0-D3/A0-A2,-(A7)
         LEA.L   FATAL_MS,A1
         MOVEA.L CHAN_1,A0
         VECTOR  UT_MTEXT,2,CALL
         MOVEM.L (A7)+,D0-D3/A0-A2
         RTE

FATAL_MS DC.W    47
         DC.B    10,'*** WARNING ***
  Unexpected exception occurred',10,0

STATICS  EQU     *
KEYTAB_A EQU     STATICS+0
CHAN_2   EQU     STATICS+4
CHAN_1   EQU     STATICS+8
CHAN_0   EQU     STATICS+12
DEFAULT  EQU     STATICS+16
ERR_STAT EQU     STATICS+20
COM_LEVL EQU     STATICS+22
COM_0    EQU     STATICS+24
COM_1    EQU     STATICS+28
COM_2    EQU     STATICS+32

EXCP_TAB EQU     STATICS+36
D_SPACE  EQU     STATICS+116
```

The saga of the Atari ST computers contines to dominate the news this month. As our exclusive analysis of the 520ST's hardware in this issue shows, there is no doubt that the computer exists – the questions centre around just when Atari will manage to get machines off their Far Eastern production lines.

It is only when the company get stocks into the shops that Atari will begin to see the cash flowing into its coffers (assuming, that is, that the machine sells).

We will not indulge in speculation in these pages, it is our hope that the STs make it to the UK in the near future – if Atari is looking for guinea pigs to experiment with the computer there will be no shortage of volunteers around the Electronics and Computing offices.

80 column screen

● While the Dragon's software-generated 51 column screen (available when the machine is operating under OS9 or FLEX) is a vast improvement on the caps only 32 column screen of the standard computer, many applications packages running under either OS assume an 80 column terminal. Anyone who has spell checked a Stylograph text file will testify to the fact that the lack of an 80 column facility can be very irritating.

As reported last month, Eurohard is working on an 80 column modification based on a new version of the 6847 VDG (visual display operator). This will be of little comfort to existing Dragon users though, and in addition it is anyone's guess as to when the new version of the computer will become available.

But Compusense look as if they will be able to come to the rescue of users who wish to upgrade to the full width screen. We cannot reveal too many details of the design – it is still at the wire wrap prototype stage – but as well as providing the enhanced screen display the mod will build in some additional memory. The bad news for devotees of OS9 is that the power of the board will only be appreciated by users of FLEX. This is because to make use of the extra board it will be necessary to amend the operating system slightly. Compusense are able to do this in the case of FLEX, they have access to an annotated assembly listing of the source code, but are prevented from making alterations to OS9 because of various licensing problems.

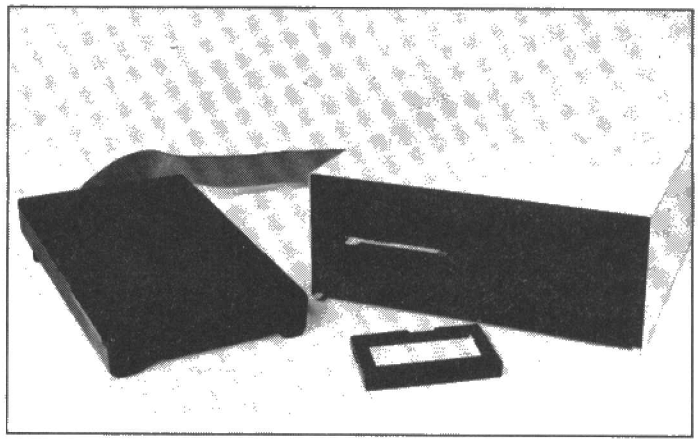
We'll have more news of the add-on in the next issue when we hope to be able to put a production unit through its paces.

Dragon expansion

● Compusense feature in another major item of news that will interest Dragon users. The company will be distributing a range of products designed to expand the basic Dragon computer (32 or 64K version).

The heart of the system is the Expansion Box that allows up to four cartridges to be connected to the computer at the same time. The standard machine only allows one cartridge to be connected at any one time. Plug in the disk controller cartridge and it is impossible to, for example, attach an EPROM blower.

Selecting between the various cartridges connected to the system is simply achieved by a POKE to location FEFF. On power up the system will default to slot 0 and will auto-boot any ROM software plugged in to this slot.



Dragon twin floppy tape unit from Compusense.

The second of the new products is an EPROM blower board. This supports a wide range of single rail EPROMs, with selection of the particular type to be programmed made via software – users are not expected to set up an array of DIL switches.

The 'blower' builds in a number of safety features that make it unlikely that a device will be damaged by the programmer, even if it is plugged in upside down.

To complement the EPROM programmer, Compusense will be distributing a sideways ROM cartridge. This allows up to four sets of EPROM software to be connected to the Dragon. Each device can be switched in via five additional BASIC commands.

A unit that should be of interest to Dragon 32 users is an RS232 interface. The design supports baud rates from 50 to 19.2K and, because it uses the serial interface, does not involve a mass of POKEs. Additional BASIC commands support most operations. In addition the unit contains a 6522 VAI providing, amongst other facilities, two 8-bit parallel I/O ports and two 16-bit programmable timers.

The best has been saved to last. The final new product from Compusense is a floppy tape unit for the Dragon. This twin drive system retails for less than the cost of a single disk drive and offers a performance that is half way between that of the computer's cassette system and a disk drive.

The operating system of the floppy tape system is compatible with that of the standard 32 computer and adds many additional commands associated with control of the twin tape drives. The drives use a digital recording technique that offers a high degree of reliability.

We hope to prepare a full review of the system in time for the next issue of *E&CM*. In the meantime, further information on the floppy drive and any other of the products mentioned can be obtained by sending a large SAE to Compusense at *PO Box 169, 286D Green Lanes, Palmers Green, LONDON, N13 5XA.*

Dragon competition winners

The 'Win a Dragon' competition, launched in our April issue, attracted a large number of entries. Four Dragons were up for grabs, the prizes going to the readers that produced the most imaginative applications for the computer. In late April Gary Evans, Editor of *E&CM*, and Ted Opyrchal, MD of Compusense who kindly donated the prizes, sifted through the mountain of entries in an effort to select the winners.

The standard of entries was uniformly high which made the task of selection all the more difficult but eventually four distinguished themselves from the rest.

The first of the Dragons goes to *Martin Hallon* who works at a residential special school for maladjusted boys. The application outlined involves using the Dragon computer, in conjunction with the RMS database software to build up a complete profile of each boy at the school.

The second winner also outlined an application with an educational bias. *Alfred Want* is in charge of a class of ten-year-olds and would use the computer both to maintain a record of the achievements of each child in his care and to present educational material in an interesting fashion.

Dragon number three goes to *Roger Evans* (no relation, Ed.) who works with one of the most powerful laser systems in Europe. His application involves using the Dragon's four A/D inputs to measure channels of laser energy.

The fourth winner is *J. R. Heath*. This application puts the Dragon into a medical environment. The analogue channels would be used to provide a real time measure of lung volume and airflow resistance. This would allow Mr Heath to use the Dragon as an intelligent monitor to record and anlayse a patient's breathing overnight without having to loose too much sleep himself.

Congratulations to all the winners and thanks to all those that entered the competition.

Anatomy of the DRAGON

Mike James concludes his dissection of the Dragon 64 with an examination of the I/O facilities.

From the electronics viewpoint, the pair of PIAs (peripheral interface adaptors) that make up the bulk of the Dragon's I/O circuits are very simple devices, but in combination with software they provide an incredible range of facilities.

It is impossible to go into the workings and programming of a standard PIA in this article and to get the most out of the explanations of how everythings works you do need to study a 6821 (or a 6820) data sheet.

The first PIA (IC5) occupies addresses FF00-FF03 and the second (IC8) FF20-FF23. As the two work together it is not possible to divide up the diagram and describe what each chip does in isolation: a better way of understanding the circuit is to see how each of the Dragon's I/O facilities is produced. Dragon I/O consists of:

- The keyboard
- The printer port
- The sound generator
- The cassette port
- The joystick ports

In addition the PIAs perform system control functions.

System control and interrupts

A number of the output lines of IC8 are used to set the display mode of the video controller chip (as described last month) these are PB3, PB4, PB5, PB6 and PB7. Interrupt input CB1 is used by a ROM cartridge to signal its presence. On detecting the cartridge the system tidies things up and jumps to the code in the cartridge ROM. The interrupt inputs – CA1 and CB1 on IC5 – are connected to video sync signals. CA1 receives line sync pulses and can therefore cause an interrupt at the end of each displayed line of the TV raster (ie about every 64us). CB1 receives frame sync pulses and so can cause an interrupt at the end of each TV frame (ie about every 1/50th of a second). The CB1 interrupt is used to provide the Basic clock timer. The Dragon 64 also uses PB2 of IC8 to enable and disable the internal ROM so that the memory can be configured as 64K RAM. In the Dragon 32 this line isn't used and is tied to +5V via a resistor.

The keyboard

The keyboard is composed of eight column wires and seven row wires; pressing any given key connects one of the columns to one of the rows. The 8 keyboard columns are connected to the B side of IC5 (PB0 to PB7) and the seven keyboard rows are connected to the A side (PA0 to PA6). To read the keyboard a single column is enabled by setting the corresponding B side line low while all the others are high. If a key in an enabled column is pressed then the row line that it connects to the column line will be pulled low and its corresponding bit in the PIA's data register will be zero (the rest will be one). The Dragon's software scans all of the keyboard columns in turn looking for keys that are pressed. All of the keyboard features such as autorepeat and roll over are provided by the way that the software handles the detection of a keypress.

The printer port

The B side of IC5, as well as scanning the keyboard, also drives the data lines of the printer port. This apparent conflict of use is no trouble in practice – normally you are either using the keyboard or the printer but not both. The data lines are buffered by a 74244 octal tri-state buffer (IC1). The data ready strobe to the printer is provided by the other PIA (IC8) using PA1, buffered and inverted through part of IC36. The printer busy and acknowledge lines are also provided by IC8 – PB0 as the busy input and CA1 as the acknowledge input. The printer takes no notice of any pulses used to scan the keyboard because the data ready strobe is not active during reading of the keyboard. Similarly the keyboard reading isn't upset by printing because the column lines are not read during the operation.

The sound generator

The Dragon's sound generator is a general 6-bit D to A convertor. The six A side output lines – PA2 to PA7 of IC8 – are connected to a CMOS 4050 buffer (IC4). The outputs of the buffer are mixed together using a resitive divider network in such a way that each successive output voltage is divided by two. Thus if the output of bit 7 is taken as 1 the output of bit 6 is 0.5, and the output of bit 5 is 0.25 – and so on. By adding these binary weighted outputs together a voltage is obtained that is proportional to the binary number represented by the state of the six lines. Using this hardware, sound generation is a matter of software generated waveforms. By sending the right sequence of six-bit numbers to the Dragon's D to A converter it is possible to create almost any sound but BASIC provides only pure sine waves.

The output of the D to A converter is connected to a four-channel analogue multiplexer 4529 (IC6). The Y input channels of this multiplexer are connected as follows:

Y0 – the D to A converter
Y1 – the cassette input
Y2 – the external cartridge sound input
Y3 – not used

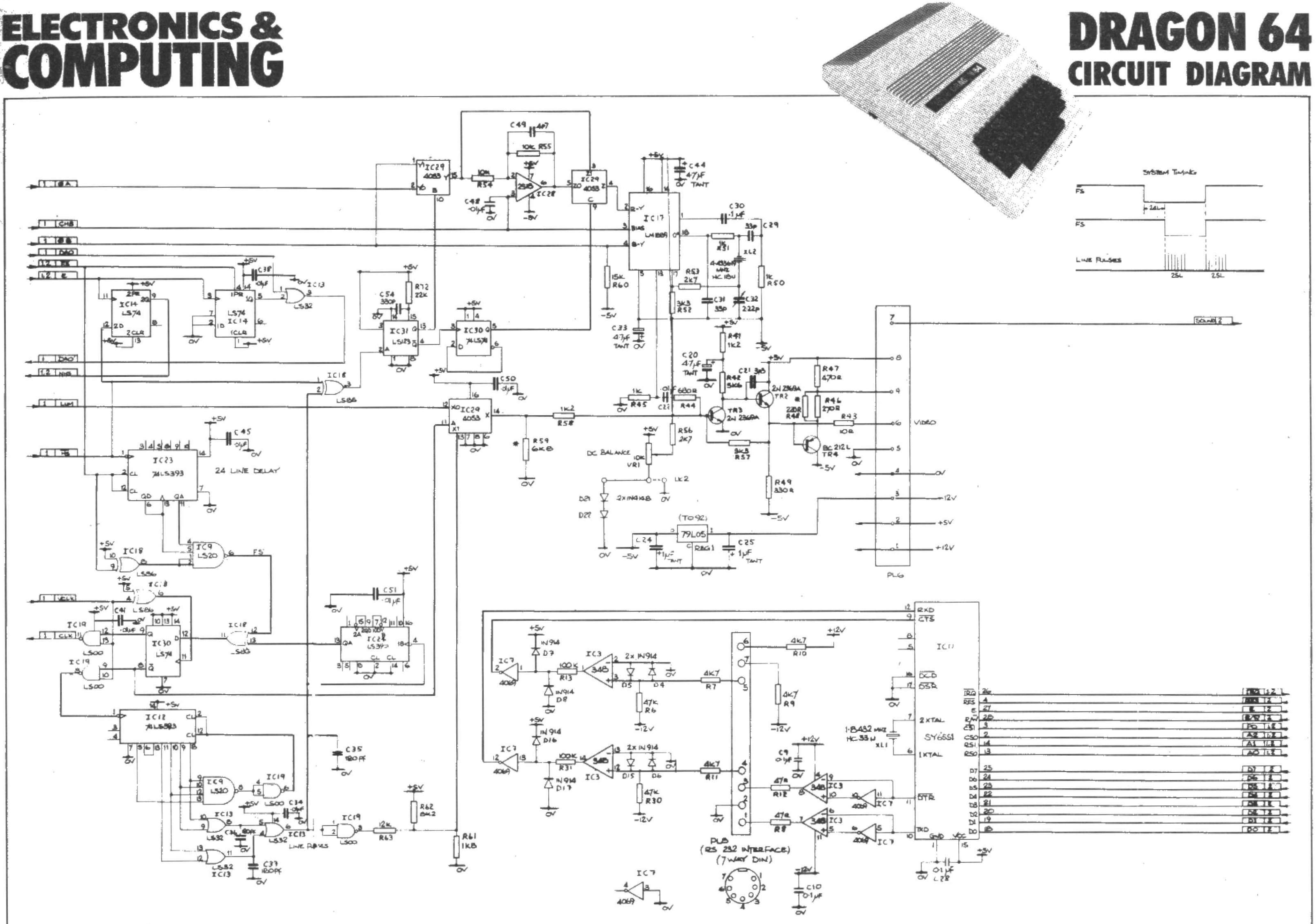
Any one of these sound inputs can be selected and routed through the TV's loudspeaker. The selection is achieved using CA2 and CB2 of IC5 and CB2 of IC8. The two inputs A and B to the multiplexer can be thought of as a 2-bit number n that selects channel Yn if STY is high. The D to A converter is used by the SOUND command but you can select the cassette sound input using the AUDIO command.

As well as the three sound sources fed to the multiplexer there is also a single bit

"... the pair of PIAs within the Dragon, in combination with some clever software, provide an incredible range of facilities ..."

ELECTRONICS & COMPUTING

DRAGON 64 CIRCUIT DIAGRAM

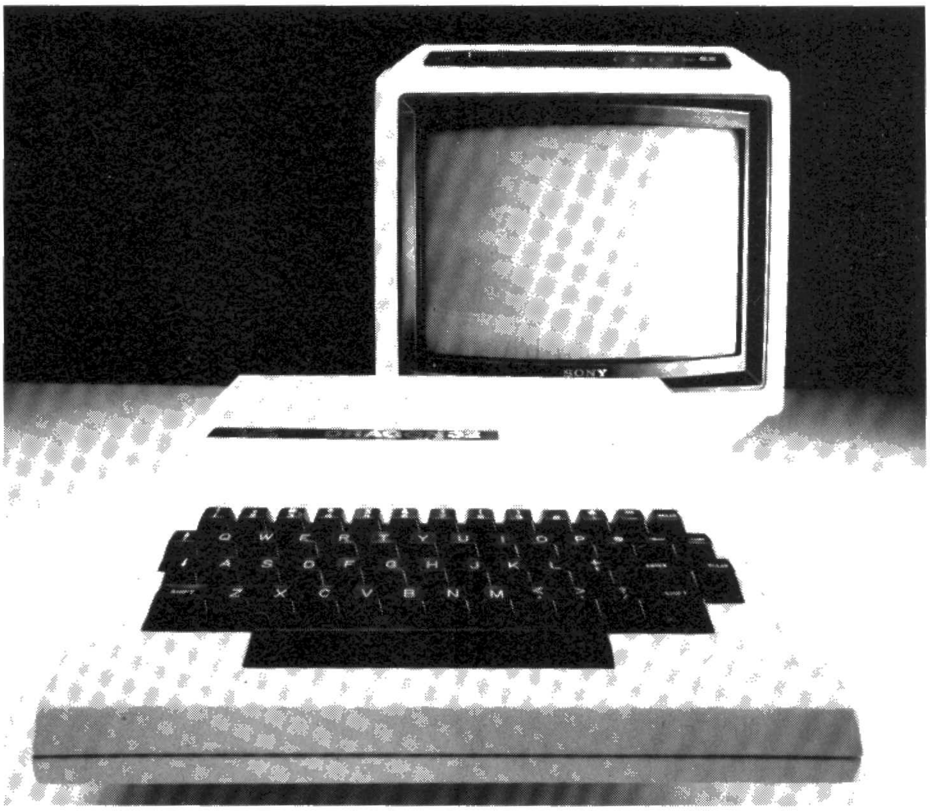


NOTE 1. * COMPONENTS FITTED FOR MK III PSU ONLY

sound source mixed in with the final sound signal by R26. The single bit sound is provided by PB1 of IC8. This can be used to produce simple buzzing noises without the need to go to the trouble of programming sine waves using the D to A. PB1 of IC8 can also be used as an input line and in this mode it acts as a sound sense line; that is, it will detect when there is any sound coming out of the TV loudspeaker.

The cassette port

The cassette port uses the D to A converter described in the previous section to generate sine waves of the appropriate frequency. These are applied directly to the tape recorder's MIC input circuit after some attenuation and passive filtering. The input circuit is almost as simple. An LM393 (IC2) acts as a zero crossing detector and feeds input pulses that are essentially squared up versions of the pulses from the tape recorder to the PA0 input of IC8. The motor control circuit is driven by the output of CA2 on IC8 via the transistor TR1 and relay RLA. The diodes in the circuit are designed to kill any transient spikes produced by the relay or the recorder's motor.



The joystick ports

The joystick ports are provided by a four-channel six-bit A to D converter. However all the parts of this apparently new component have already been described – the method of conversion is to generate a voltage using the 6-bit A to D convertor and compare it to the input voltage; if they are equal or nearly equal then we have the binary number corresponding to the input voltage and if they are not equal, we try another voltage.

The output of the D to A converter is compared to the input voltage using as LM393 comparator – this is in fact the other half of IC2, used in the tape circuit. The output of this comparator indicates whether the input voltage is bigger or smaller than the D to A voltage and is fed to PA7 on IC5. Which of the four A to D channels is connected to the comparator is determined by the analogue multiplexer IC26. This is the same multiplexer used by the sound generator and it is controlled by the same output lines of IC5 and IC8. The joystick fire buttons are simply additional connections across the PA1 and PA0 keyboard row inputs and as such have the same effect as pressing any key in either of the two rows.

The Dragon 64 serial port

One of the major differences between the Dragon 32 and 64 is that the 64 has a general purpose serial port. It consists of a single SY6551 combined ACIA and baud rate generator (IC11). It is selected using the Y2 output of IC40 (see last month) but fortunately doesn't occupy the same area of memory as IC5 because it uses address line A2 as an additional select, making its starting address FF04. Its outputs and inputs are all buffered by a 348 quad op amp IC3 and a 4069 CMOS buffer IC7. The op amp converts the signals to RS232 levels of +12V and –12V.

Fault finding

This completes the description of the circuit diagram. The only sections that have not been described are the power supply, the UHF modulator, and the final stages of video processing. However a few hints on finding Dragon faults might be useful to anytone with a dead or sick Dragon on their hands.

If your Dragon is dead then the first thing to check is the power supply. On the Dragon 32 this is simply +5V to all the logic, +12V to TR1 and –5V to the modulator. The Dragon 64 has a –12V supply to the ACIA and associated chips for the RS232 level. (Note on the Dragon 32 the 16K dynamic RAMs use +5, –5 and +12V supplies whereas on the 64 only a single +5V is required.) If the voltages are all OK and the Dragon is still dead then it is likely to be a video problem.

Video problems can generally be tracked down to faulty RAM chips but occasionally the SAM or the video generator will have failed. If you find that you cannot select one or more video modes correctly – but everything else is OK – then suspect PIA IC8. If some characters are displayed incorrectly then the data latch IC20 may be at fault. RAM problems often show up as changing patterns on the video screen and so it is worth running a RAM test program before pulling the video chain apart. Failure of the SAM chip can effect the video display but it usually shows up as a RAM fault that doesn't go away when the chip is changed, or as randomly changing bits in memory.

As all of the I/O circuits share components it is often possible to track down an error simply by noting what is still working. For example, if the joysticks fail to work but the sound generator is working and you can select the cassette using AUDIO ON then it is likely to be the comparator IC22. The only part of a chip unique to the joystick port is the comparator. Both the sound generator and the joystick port use the D to A converter, and the audio multiplexer is shared with the sound channel and the joystick inputs. It is worth mentioning that most joystick faults are in the joysticks themselves or the cables that connect them. The printe buffer IC1 is also vulnerable to attack by certain printers that have high DC voltages as part of their input connection – if a printer doesn't work check IC2 first then IC8.

In general, the Dragon is an easy beast to diagnose and repair but sometimes it is better to leave things to the experts. If the fault lies in one of the socketed chips then find out if you have to unsolder one of the 40 pin devices you will have to replace it at some stage, and stand a good chance of damaging the printed circuit board, just to make matters worse!

References:

"Anatomy of the Dragon", Mike James, Sigma Technical Press, 1983. This provides a comprehensive guide to the Dragon's hardware and the way that it interrelates with software. It is available from the *E&CM* Book Service.

Acknowledgement

Thanks are extended to Compusense who supplied the diagram used in the preparation of this article. Readers who have requirements for Dragon hardware or software are advised to contact Compusense.

The QL industry

who's who in software

Even 'super computers' depend on well written software if they are to achieve any degree of market success. The Sinclair QL has been generally available for about one year: during this time a small, select band of software houses have released software for the machine.

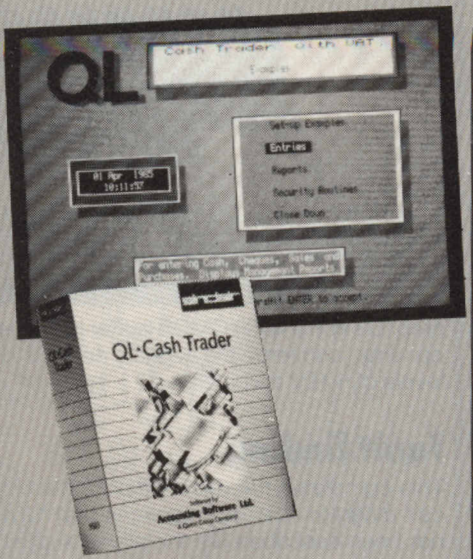
Sir Clive Sinclair, while defending the QL at a press briefing earlier in the year, claimed that his computer would see off the challenge from the likes of the Atari 520ST by virtue of the fact that the numerous applications programs available would not, at least in the short term, be matched by any new arrival. In this assertion Sir Clive is quite correct. The QL, because of its bundled software, is capable of use from first switch on.

Contrast that to the limited software likely to be supplied with the ST computers. GEM on its own is nothing to shout about and DR LOGO and a Microsoft BASIC clone are similarly not likely to set the programming world alight.

As the quality and amount of software support is acknowledged as being crucial to the success of a computer, we thought it time to take a critical eye at the products available for the QL. What follows is a survey of a selection of the software available, not a series of detailed reviews. We have cast the net as wide as possible and detail not only assemblers and tool kits but also take a look at some of the games that have been written for the Sinclair supermicro – even the most serious of users must at some stage need to ease tension.

The QL is the first of a new generation of home/business micros. It is based on the 16/32-bit 68000 processor, and so has cast off the shackles of the 64K barrier that restricted so many 8-bit micros. Speed and shear power are also far superior to the current crop of micro-computers. In some respects the power of the QL has presented a problem for many software authors. The wealth of Z80 and 6502 programming experience that has been built up over the past few years is of little use in the context of the QL. Programmers have had to get to grips with the art of 68000 machine code programming – some have taken to it easier than others and many are still not far from the bottom of the learning curve.

What has tended to happen is that



would-be software authors have tried to convert popular games, utilities and other programs from machines such as the Spectrum and Commodore 64. This was bound to fail not only because of the different screen architectures of both these computers, but also because the 16/32-bit architecture of the QL is a very different prospect for programmers.

There has yet to emerge any significant body of software which does full justice to the QL. That is not to say there is no reasonable software about, just that with the possible exception of the Psion packages, the full potential of the QL has yet to be explored. What has happened is that there is now a host of commercial programs and games to cater for a vast, seemingly undemanding public. This is proved by the plethora of small ads to be found in various computer journals. It is all very reminiscent of the early days of 8-bit programming when the ZX81 was the wonder machine and the current generation of programmers were cutting their teeth on this piece of micro computer history.

So having set out the QL software scene we will start our survey of some of the QL specific software available today; from short games to more complex adventures and professional utilities from the likes of Talent and Digital Precision.

John Banks looks at a selection of software now available for the QL.

We would like to emphasise that the following QL software index does not claim to be comprehensive – it lists those companies who have brought their products to our attention. Nor does the fact that many of the companies listed below do not have their software mentioned in the commented section above imply that there products are in any way less important, it is simply that we do not have space to comment on every product range.

CASH TRADER

Sinclair Research
Price: £69.95

According to the promotional material Cash Trader was designed for the one-fingered businessman (this presumably applys to his typing skill – first the small businessman now the one one-fingered) with virtually no accounting skill, who is looking for instant results and accurate information. On most counts Cash Trader succeeds.

That some 300K of code is spread across three microdrives may indicate that the program is difficult to use. Not so. The manual provides a host of worked examples introducing the user to the concepts of book keeping. Coverage is comprehensive and easy to understand.

Cash Trader is targeted at the sole trader where business is mainly on a cash basis and where records tend to be hapazard. It finds its mark precisely. As the first piece of true business software for the QL it is likely to take the market by storm – assuming the aforesaid businessman owns a QL. The quality of the product is indicated by the fact that it is currently being ported across to very much more expensive desktop computers. If you happen to be a small, one-fingered business man, do try and have Cash Trader demonstrated for you.

QL APPLICATIONS SOFTWARE

Company	Contact	Products
Sinclair Research	0376 686100	QL Cash Trader, QL Entrepreneur, QL Project Planner, QL Decision Maker,
Cenprime	0203 686162	QL Bank Account
Adder	0223 277050	Q-Doctor, Assembler
Computer One	0223 86216	Pascal, Forth, Assembler, Monitor, Typing Tutor
Co-op Soft	0272 22223	Civil/Structural Engineering
Dialog Software	01 500 2386	Transact Book-keeping package
Digital Precision	01 527 5493	Games Designer, Sprite Generator, Dissassembler + Monitor
GST	0954 81991	Q Jump, Toolkit, Assembler
Harcourt	0276 686100	Touch 'n go
Hisoft	0582 696421	Mon-QL
Metacomco	0272 428781	Assembler, BCPL, Lisp
MicroAPL	01 622 0395	
Portfolio	PO Box 15, LONDON, SW11	Stockmarket Manager
Positron	0554 759624	Hi-res screen dump
Q-Soft	01 449 7417	Agenda Desk Diary
Quest	04215 66488	Business accounts
Super Plant	097 423223	Shrub Planner, House plant planner
Tasman	0532 438301	Tascopy Screen Copier
TDI Software	0272 742796	USCD Pascal, USCD Fortran, Advanced Toolkit
TR Systems	093 924 621	QL payroll

QL GAMES

Company	Contact	Products
Digital Precision	01 527 5494	Backgammon
Eidersoft	01 478 1291	QL Art
Games Workshop	01 965 3713	D-Day
Printerland	0484 513105	Psion Chess
Shadow Games	0296 686100	Area Radar Controller
Talent	041 552 2128	ZKUL, WEST, GraphiQL

PROJECT PLANNER

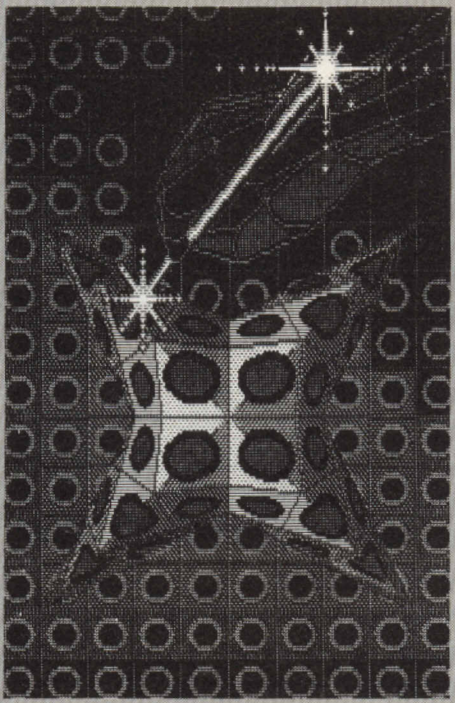
Sinclair Research

Price: £39.95

Project Planner is an easy to use package in which the user is encouraged to break down a task into a number of component jobs. Each job is given a description, duration and cost to complete. When any one job is dependant on others having been completed the user is required to specify them. It is at this stage that the program comes into its own.

By applying the theory of Critical Path Analysis the program creates a horizontal bar chart showing the overall duration of the project, and highlights those tasks that are critical and those where a degree of leeway in time of completion is permisable. This allows the user to determine 'at a glance' those areas where a day or two saved will offer most benefit.

The program is undemanding in terms of the data required to be entered at the keyboard and, providing that the overall task can be split into a number of jobs, this is a useful addition to the tools of a manager.



GRAPHIQL

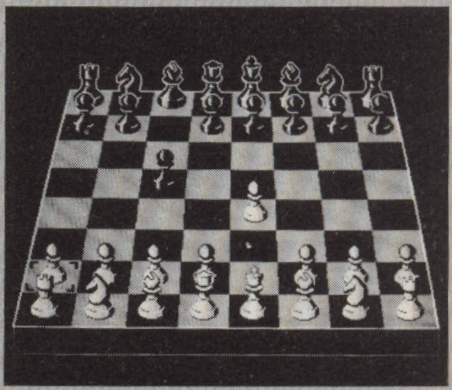
Talent Computer Systems

Price: £34.95

Graphiql is aimed at the artist and recreational user as opposed to the draughtsman or engineer. Using the QL's low-resolution screen, the emphasis is on colour and composition rather than on pin point accuracy. With features such as magnification and texture definition however, it is a considerable advance upon an electronic sketch pad. Not quite Computer Aided Design, but certainly a step in the right direction.

Talent Computer Systems, Curran Building, 101 St. James Road, Glasgow, G4 0N8. Telephone: (041) 552 2128.

QL Chess from Psion



TASCOPY

Tasman Software

Price:

The Tascopy program allows grey scale, monochrome and poster sized screen copies to be produced on a dot matrix printer that supports a bit image mode. A window facility allows the user to select any area of the screen for printing.

Another option allows the user to specify how any particular colour will be represented by the printer.

Tascopy has been tested with a wide range of popular printers and serves the very useful function of producing high resolution screen copies of the QL's screen display.

Tasman Software, Springfield House, Hyde Terrace, Leeds, LS2 9LN. Telephone: (0532) 438301).

SUPER SPRITE GENERATOR

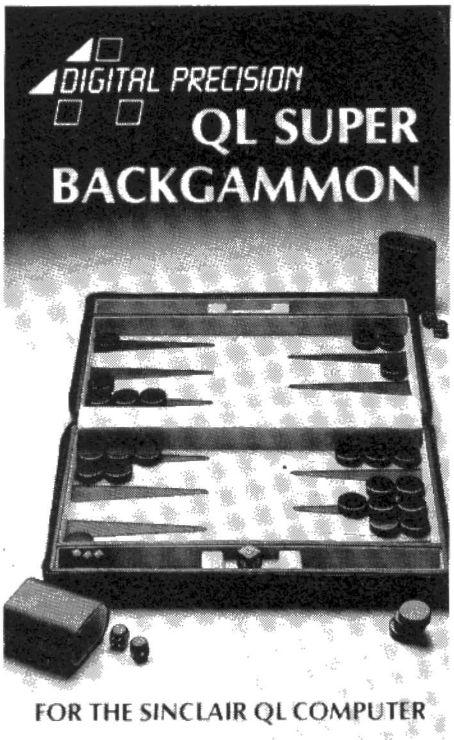
Digital Precision

Price: £24.95

This package is designed to capture the interest of users who want to make the most of the superb graphics capabilities of the QL. The program draws a grid on the screen and using the cursor keys, the user can move around the screen defining a sprite.

Once finished the sprite pattern can be saved. An animated display can be produced by designing a series of sprites, each one a progression in movement from the last. When the desired movement sequence is obtained, the sequence of frames is put into a single file which can be called by the user's own programs via SuperBASIC extensions. These activate, control and animate the sprites. This process is the only difficult part of the program as the functions relating to movement, collisions, planes etc have quite complex parameters. It's worth putting in some effort to get used to them however, since the results that can be achieved are excellent.

Digital Precision, 91 Manor Road, Highhan Hill, LONDON, E17.



QL TOOLKIT

Sinclair Research
Price: £24.95

The Toolkit takes over from where SuperBASIC leaves off. The extensions to SuperBASIC are contained in a 9K machine code file (EXTEN_BIN) and are loaded when the package is first booted up.

A total of 58 or so additions are provided – everything from simple file maintenance commands such as RENAME and TRUNCATE to more complex commands associated with random access of files.

The Toolkit will have little appeal to the novice programmer, as extended SuperBASIC is still hard to master. For the more advanced programmer the package should be most welcome: that's assuming they have not coded many of the extensions themselves.

Sinclair Research, Stanhope Road, Camberley, Surrey. Telephone (0276) 686100.

ASSEMBLER

Computer One
Price: £29.95

Computer One's assembler puts the emphasis on ease of use and speed. The absence of macros and a linker are unlikely to affect all but the most advanced programmers. Concurrency, the ability to edit and assemble simultaneously more than makes up for minor shortcomings in the manual and screen handling, and compensates for the limited number of directives available. This gives a professional edge to what is a modest priced product.

Computer One, Science Park, Milton Road, Cambridge. Telephone (0223) 862616.

D-DAY

Games Workshop
Price: £24.95

D-Day features some 200K of programming, with about 20K devoted to computer intelligence.

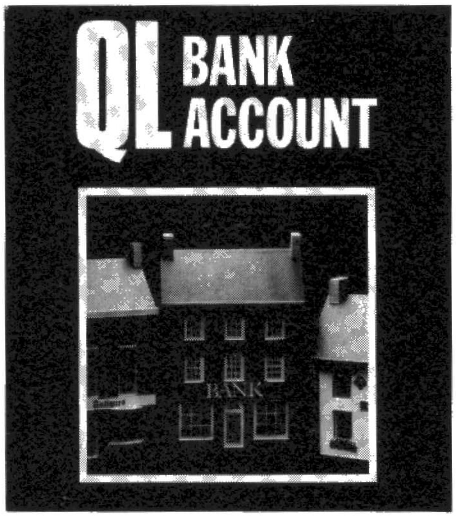
On loading the player is presented with a series of menus, the first one offering the choice of a one or two player game.

There is a choice of senario – four are provided – and the player can opt either for large or small forces.

The screen is divided into two windows, the upper displays a tactical map and the lower is used to display available instructions.

The mechanics of play are straightforward and the graphics certainly impressive.

The programming of the game is more than adequate although a little more code devoted to computer intelligence would be welcome. As it stands the program is able to provide a fairly strong non-human opponent but one can't help but feel that much more could have been done with the available memory. As we pointed out in the introduction to this article, this is a failing of so much QL software.



TYCOON

Newtech Publishing
Price: £19.95

This game combines crosswords with a business simulation; a combination that provides an ingenious result. Up to six players have to make as much money as possible – they do this by solving the crosswords and selling the solutions to the bank.

There are 49 crosswords, graded by difficulty. Each player may select their own level or let the computer select one at random. Options during play include buying random or specific letters which are added to the grid, taking or repaying a loan, investing, or selling a word.

Tycoon, with its original conception, is a fascinating addition to the library of QL software.

Newtech Publishing, 8 Ferge Court, Reading Road, Yateley, Camberley, Surrey.

SUPER BACKGAMMON

Digital Precision
Price: £15.95

This is an implementation of the increasingly popular boardgame, and a good one at that. The game is for two players, in the case of Super Backgammon this can either be between two human players, the computer being used as an 'intelligent' playing board, or between you and the computer. Be warned that in the latter case the computer will almost certainly come out the victor.

The program includes full instructions, both on the packaged insert and on a useful help page.

Moves are registered simply – the computer will ignore any illegal moves. 100 can be entered either if there is no legal move or if you require a suggested move from the QL.

The attention to programming detail evident throughout Super Backgammon results in a package that is very enjoyable to use.

Digital Precision, 91 Manor Road, Higham Hill, LONDON E17.

MONITOR

Computer One
Price: £24.95

The Computer One Monitor is much more than an aid to the debugging of a machine language program. In addition to the debugging function, it monitors just about every aspect of the QL at a useful, low level.

The manual supplied with the Monitor is over 40 pages long and is packed with a mass of useful information.

The overriding impression of this product is of an invaluable addition to any machine code programmers tool kit. The Monitor seems well worth the outlay.

Computer One, Science Park, Milton Road, Cambridge. Telephone: (0223) 862616).

POSTSCRIPT

Care Electronics has pointed out that it can supply a range of 85 character monitor/tv sets designed for use with the QL. Unfortunately the company was not listed among monitor suppliers in our QL Hardware round up published last month.

Care has also asked us to point out that while discussing serial to parallel converters for use with the QL it was suggested that the most sensible option was to purchase a low cost fixed baud rate design. While such converters are perfectly adequate if users only wish to interface the computer to a printer, a fixed baud rate design will offer less flexibility if it is necessary to interface the QL to other items of equipment that do not support the QL default rate of 9600. The Care interface provides a variable baud rate facility to cope with such situations.

THE FIRST SPECTRUM WORDPROCESSOR IN FIRMWARE PART FOUR

THE 2K TEST

The software and operating system for Richard Sargent's Spectrum wordprocessor is held on an 8K EPROM, but you can have a 2K RAM based wordprocessor without building or burning anything.

The wordprocessor is the what-you-see-is-what-you-get variety, which means that you can format the text on the screen, see what it looks like, change it if necessary, and then get a perfect print-out.

But in reality the Spectrum, in common with many other computers, can't achieve perfect WYSIWYG because the screen width is nowhere near the 132 characters offered by most printers. So, reality on the horizontal plane is, unfortunately, not possible. Instead, the lines wraparound the screen for any line-width setting which is greater than 32 – the width of the Spectrum screen. This might be a disappointment, but unless you want to use a wordprocessor for work involving columns, it doesn't matter in the slightest. Line beginnings, paragraph layout and the effect of various line-width settings are all shown clearly. Wrap-round on the screen may lead you to suppose that you've typed a document which is very long, so a line counter is provided to assure you that this is not necessarily the case!

The RAM-based program is just under 2K long and represents the basic operating system destined to become the heart of the EPROM. 2K *is* a lot to type in, but there is no easy solution.

As soon as the machine code shrinks in size towards 1K, the supporting BASIC program grows correspondingly as it takes on more and more of the work. By contrast, the 2K program is driven by about 18 lines of BASIC. A breakdown of the code shows the routines which are greedy for RAM, and as you might expect after last month's excursion into the workings of the Spectrum screen, the VDU routine looms large with 100H bytes. The figures in **Table 1** are rounded off, but they serve to show how quickly the code mounts up. Taking these routines as a guide, the workings of the wordprocessor can be understood.

TABLE 1

INITIALISATION	100H
COMMAND TABLE	070H
MAIN LOOP	040H
VDU ROUTINES	100H
KEYBOARD	038H
CURSOR CONTROL	0B8H
UPDATE-VALUES	118H
INSERT/DELETE	118H
FORMAT LINES	0B8H
PRINTER DRIVER	078H
SUNDRIES	100H

	800H

Booting up

The initialisation is in two sections: COLD START and WARM START. Warm start is the entry point to the wordprocessor and is used whenever you wish to preserve the document currently in RAM. The code does very little other than erase the screen contents (which might be a BASIC direct command) and to lay out a new screen in wordprocessor format. Cold start, on the other hand, performs numerous preparatory tasks, such as setting a fast key-repeat rate, turning off caps-lock, and initialising the printer's PIO. It also moves the Spectrum's character-set into RAM, for reasons which will become clear later. The document file is set to one byte, and this contains a space character over which the cursor blinks.

Command table

This table holds the 32 command letters recognised by this shortened version of the wordprocessor. Each command has a sub-routine associated with it, with its address also held in the table. This table can be modified (even – by devious means – in the EPROM version) so that users can add their own commands into the system.

Main loop

The wordprocessor spends a fair proportion of its time in this loop, waiting for the user to press a key. While waiting, it performs various useful tasks, such as refreshing the screen with text from the document and updating the line counter and memory-available indicator. It also supervises the only part of the wordprocessor screen (lines 2 and 3) which is allowed to scroll in the computer sense ie: when line 3 is full, the information on it is scrolled onto line 2. Line 3 usually holds a message relating to the command being executed, and when a new command is issued it too goes to line 3, with the previous message passing up to line 2. It follows that you can usually see what you're doing (or rather what the computer thinks you're doing) and also what you've just done, and this can be quite useful if you're in the habit of pressing the odd key or two by mistake.

VDU routines

There are four main routines here, and the first, the direct screen-print routine (which doesn't fuss about channels, attributes or print-controls) was discussed last month. The second routine is the screen-refresh routine and it uses the direct screen-print routine. Its job is to display a 17 x 32 character portion of whatever text is in main memory, and can therefore be regarded as a viewing window looking at your document. This window must be refreshed after every keypress (character insertion, deletion or cursor movement) and it is here that there are potential speed problems, since no matter how many letters appear on-screen, all 544 character cells of the window are refreshed. Whenever the 8 bytes of a single character cell is refreshed, information is extracted from the character shape-table and this takes time. However, a screen window will quite often contain blank lines, and these are refreshed more quickly than lines containing characters: a special subroutine

TABLE 2

FA00	00000000
FA01	01000100
FA02	10101010
FA03	10101010
FA04	10101010
FA05	01000100
FA06	00000000
FA07	00000000

outputs trailing spaces without the need to look up space-code 20H in the shape-table.

The shape table used by the wordprocessor is at FA00H, and is a copy of the Spectrum's own character set, with special symbols for codes 00H to 1FH added. The symbol for code 00H, for example, occupies addresses FA00-FA07 and appears on the screen as two neat little zeros. **Table 2** shows how the shape-table is built, and, since it's now in RAM, you can alter some or all(!) of the character shapes if you want to.

The screen-refresh routine has a special "easy read" mode which serves two purposes. Pressing caps-shift and symbol-shift together at any time toggles the screen from a 544-character screen to a 256-character screen. This is not only useful to users with poor eyesight or a cheap television set, since this window is refreshed twice as quickly as the other type and permits faster scrolling.

LISTING 1

```
;ENTER WITH ROW NUMBER IN B
;AND COLUMN NUMBER IN C
;ie BC=00 IS TOP LEFT OF SCREEN

INVDU   PUSH HL
        PUSH DE
        PUSH BC
        LD DE,NEWUDG
        LD A,B
        AND 18H
        ADD A,40H
        LD H,A
        LD A,B
        AND 7
        RRCA
        RRCA
        RRCA
        ADD A,C
        LD L,A
        LD B,8
LOOP    LD A,(HL)
        LD (DE),A
        INC H
        INC DE
        DJNZ LOOP
        POP BC
        POP DE
        POP HL
        RET
NEWUDG  DEFS 8

;EXITS WITH THE CHARACTER
;STORED AS A MINI-UDG TABLE
;IN NEWUDG
```

The third VDU routine reads data from the screen, and is needed because the type of cursor chosen (an arrow) alternates with the text-character "underneath" the arrow. This method is almost certainly slower than those which provide a cursor by flashing the text-character between normal and inverse video, but I prefer the arrow! The routine may be of general interest since it is quite short, and it's shown in **Listing 1.**

The fourth VDU routine prints messages on the top four rows of the screen. To save some space the Spectrum ROM routine PRINT-A-2 at 15F2H is used in conjunction with the message reading routine at 203CH. The ROM's CLEAR SCREEN (0D6BH), CLEAR LINE (0E44H) and SET COORDINATES (0DD9H) are also used to support this fourth VDU routine. If you have *The Spectrum ROM Disassembly* by Ian Logan and Frank O'Hara, you can look these routines up. They are most useful, but you should be aware of one neat little Sinclair trap for the unwary. In BASIC the text co-ordinates count across and down the screen, hence it's sensible to suppose that when these co-ordinates are tucked into the BC register (C for columns), BC=00 will represent the top left hand corner of the screen. However, when you use ROM routines, this corner is known as B=18H, C=21H, because the numbers are calculated as B=18-row and C=21-col. **Listing 2** shows how these routines can be used in programs of your own. Notice that channel S, the screen channel, is opened at the start of the routine, in case some other routine has switched output to a different channel.

LISTING 2

```
;WRITE A STRING TO THE SCREEN

TEST
LD A,2
CALL 1601H ;OPEN SCREEN CHANNEL
LD B,18H
CALL 0E44H ;CLEAR 24 ROWS (ie:CLS)
LD B,0DH
LD C,1EH   ;0D/1E IS MID-SCREEN
CALL 0DD9H ;SET THOSE COORDINATES
LD DE,1539H;ADDRESS OF STRING
LD BC,19H  ;LENGTH OF STRING
CALL 203CH ;PRINT STRING
RET
```

Readers with the ROM Disassembly may also wish to investigate the print-to-channel routines generally, which start in earnest at 0B24H and run on to about 0C41H, and the number-printing routines at 1A1BH, 1A28H and 1A2CH. 1A28H and 1A2CH seem useful, but they caused an early version of the wordprocessor to crash and are out of favour in certain quarters.

Extra keys

The original Spectrum never had enough keys and the advent of Spectrum Plus has made real typing possible, but although the physical number of key-positions has increased, the real number of electrical switches scanned by the software remains the same. The hardware of the Spectrum wordprocessor puts this to rights by providing auxillary switches, but in the meantime there are some useful ways of handling the Spectrum keyboard which permit fairly effortless wordprocessing.

The RAM-based wordprocessor could, for example, accept up to 72 commands using the 36 separate keys and their shifted counterparts. The actual commands provided are shown in **Table 3.** But this is the COMMAND-MODE in operation; the problems start when text is being inserted into the document and then very few keys are available for special functions. The culprit is the keyboard-read routine, based on LD A,(23560), which simply reads the last keypress obtained by the ROM routine at 02BFH.

Although this method won't return codes 00-1FH, 5BH-5EH, and 7BH-FFH it does not cause a total disaster; codes 80H-FFH are not required anyway, and codes 5BH-5EH, 7BH-7FH are fairly specialised. Codes 00-1FH, on the other hand, are required in documents to control printers and the program allows them to be entered into the text by the simple expedient of typing <shift-4>, <n>, <n> where nn is the hex number for the code you require. To set an Epson to print in emphasised mode, you would type <shift-4>,1,b,E. The letter element of the code 1BH must be in lower case. <1BH>,<E> is the full code for starting emphasised print, and it will show on the screen as 1BE, with the 1B occupying just one character square.

TABLE 3

```
THE COMMANDS AVAILABLE FOR RAM-W.P.

I INSERT MODE
i INSERT SINGLE CHARACTER
c CHANGE SINGLE CHARACTER
d DELETE SINGLE CHARACTER
D DELETE A LINE
F FORMAT LINES
W WIDTH OF LINE
B RETURN TO BASIC
P PRINT
E ERASE DOCUMENT
H HELP
G (GET) LOAD FROM TAPE
M MERGE A DOCUMENT
V VERIFY
S SAVE TO TAPE
C COUNT WORDS
U UPDATE COUNTERS AUTOMATICALLY

PLUS 13 different cursor/scroll commands.

INSERT MODE supports delete and
    4 cursor movements.
```

Having used shifted-4 to embed control characters, the other shifted numbers can now be allocated tasks. Four become the cursor arrows, one is DELETE, and pressing shifted-2 produces the CAPS-LOCK code. That leaves just three: shifted-1 (EDIT) is used to escape from INSERT MODE; shifted-3 and 9 are used for the important task of marking blocks of text which need to be immune from the formatting process. Shifted-3 marks the start of the block with the square bracket (code 5BH) and 9 marks the end with the other square bracket (code 5DH). Wordprocessors need a TABULATE key, and symbol-shifted-0 is used for this purpose, which means that the symbol involved, the underscore line, cannot be entered into your text.

Cursor control

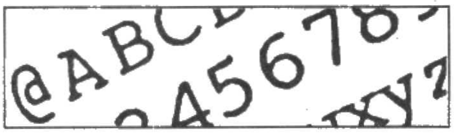
Thirteen cursor control and scroll commands are supported in the command mode – 8 on the arrows keys, two on ENTER and EDIT, and three more on the Q, A and Z keys.

The status lines at the top of the screen indicate the number of free bytes left in RAM, the number of words written, the total number of lines in the document, and the number of the line at the cursor. Calculating these values and printing them

(using PRINT-A-2) takes time, and so they can be switched off by the toggle key in command mode. A single byte of RAM can hold 8 "software switches" and toggling an individual bit-switch is straightforward using the Z80 XOR instruction. **Listing 3** shows how this principle is used to toggle the CAPS LOCK switch, which is BIT 3 of the Spectrum's FLAGS2.

LISTING 3

```
LD A,(23658)  ;examine all 8 switches
XOR 8         ;toggle bit 3
LD (23658),A  ;record the new status
RET           ;job done
```



Format lines

The on-screen formatting is confined to line length in the RAM-based wordprocessor. The system is not automatic, and lines will not be set to their correct length unless you deliberately type a carriage return, or use the F command. It follows therefore that if you just keep typing, a line can grow in size and keep on growing until the document is finished. If this is done, the automatic windowing which follows your progress as more text is typed will sometimes fail to keep up. The text will still be entered into memory, but you won't see it. The occasional paragraph break (two carriage returns) and/or the use of the F command will put things right. If you type a *single* carriage return, it will not survive the formatting process unless you have protected the particular piece of text concerned with the shift-3 and shift-9 markers.

The cause of this lies in the formatting algorithm, which essentially treats single spaces and single carriage returns as interchangeable items. In this way text can be repeatedly formatted. Work on the monitor screen is best done with the line length set to 32, and for checking and at print-time it can be reset to a more suitable width.

A side benefit of this procedure is simple document text with no visible line markers, such as 0AH, 8DH or 8AH. Unless you deliberately enter control-code bytes, 0D11 is the only byte outside the range 20H-70H which will be found in the text. Code FFH marks the start and end of the text, but generally characters with codes in the range 80H-FFH cannot be held in the text since the routine used to calculate the on-screen cursor position sets BIT 7 as a marker, and would be confused by the presence of graphics in the text!

LISTING 4

```
THE BASIC LISTING

   1 LET n$="": REM ram version 16-4-85
   2 GO SUB 9: LET a=USR 32771: GO TO 10
   5 LET n$="": GO SUB 9: LET a=USR 32768: GO TO 10
   6 GO SUB 9: LET a=USR 32768: GO TO 10
   9 BRIGHT 0: PAPER 7: INK 0: BORDER 5: RETURN
  10 IF a=0 THEN  STOP
  12 IF a>10 THEN  LET a$="SAVE": GO SUB 50: SAVE n$CODE 34842,a: GO TO 2
  14 IF a=1 THEN  LET a$="LOAD": GO SUB 50: LOAD n$CODE : GO TO 2
  16 IF a=2 THEN  LET a$="VERIFY": GO SUB 50: VERIFY n$CODE : GO TO 2
  18 IF a=3 THEN  PRINT AT 2,1;"F-NAME TO BE MERGED:": INPUT b$: PRINT AT 3,1;b$
: LOAD b$CODE : GO SUB 9: LET a=USR 32774: GO TO 10
  20 IF a=4 THEN  GO TO 100
  22 IF a=5 THEN  GO TO 80
  24 STOP
  50 PRINT AT 3,1;a$;" F-NAME:";n$: INPUT "<ENTER> or <name>:";c$: IF LEN c$>0 T
HEN  LET n$=c$
  56 PRINT AT 3,1;a$;" F-NAME:";n$: RETURN
  80 INPUT "WIDTH 10-255 ";w: IF w<10  OR w>255 THEN  GO TO 80
  82 POKE 34855,INT (w): GO TO 2
 100 PRINT AT 3,1;"HELP FILE NOT LOADED!": GO TO 2
9000 SAVE "swp" LINE 9999: SAVE "swp"CODE 32768,2048: SAVE "chset"CODE 64000,256
: STOP
9999 CLEAR 32767: LOAD "swp"CODE : LOAD "chset"CODE : GO TO 5
```

SHAPE TABLE

FA00	00	44	AA	AA	AA	44	00	00	FA80	00	44	CA	4A	4A	E4	00	00
FA08	00	44	AC	A4	A4	4E	00	00	FA88	00	44	CC	44	44	EE	00	00
FA10	00	4E	A2	AE	A8	4E	00	00	FA90	00	4E	C2	4E	48	EE	00	00
FA18	00	4E	A2	AE	A2	4E	00	00	FA98	00	4E	C2	4E	42	EE	00	00
FA20	00	48	AA	AE	A2	42	00	00	FAA0	00	48	CA	4E	42	E2	00	00
FA28	00	4E	A8	AE	A2	4E	00	00	FAA8	00	4E	C8	4E	42	EE	00	00
FA30	00	48	A8	AE	AA	4E	00	00	FAB0	00	48	C8	4E	4A	EE	00	00
FA38	00	4E	A2	A2	A2	42	00	00	FAB8	00	4E	C2	42	42	E2	00	00
FA40	00	4E	AA	AE	AA	4E	00	00	FAC0	00	4E	CA	4E	4A	EE	00	00
FA48	00	4E	AA	AE	A2	42	00	00	FAC8	00	4E	CA	4E	42	E2	00	00
FA50	00	4E	AA	AE	AA	4A	00	00	FAD0	00	4E	CA	4E	4A	EA	00	00
FA58	00	4C	AA	AC	AA	4C	00	00	FAD8	00	4C	CA	4C	4A	EC	00	00
FA60	00	4E	A8	A8	A8	4E	00	00	FAE0	00	4E	C8	48	48	EE	00	00
FA68	00	4C	AA	AA	AA	4C	00	00	FAE8	00	4C	CA	4A	4A	EC	00	00
FA70	00	4E	A8	AE	A8	4E	00	00	FAF0	00	4E	C8	4E	48	EE	00	00
FA78	00	4E	A8	AE	A8	48	00	00	FAF8	00	4E	C8	4E	48	E8	00	00

Print-out

The ZX printer is not supported directly by either the RAM or the EPROM software. If anyone wishes to use this printer for wordprocessor applications, then I would suggest that the text file is PEEKed from BASIC, and each byte sent to the ZX printer using LPRINT CHR$(x); the start of the file is at 882CH=34860 and it ends when the PEEK returns FFH (255). The EPROM hardware supplies a Centronics port, but the printer routine is addressed via a jump byte in RAM and so can always be changed to suit a user's own requirements.

The printer code in the 2K HEX-DUMP is set for the same type of PIO interface as that shown in the circuit diagram of last month, and the bytes which send the program flow to the printer are the C3 0C 80 at 8828H=34856. This jump may be changed to point to a commercial Centronics interface package. Alternatively, you can write your own printer routine between 800CH and 8037H. Preserve all registers used, and end the routine with RET. Printing starts at the cursor and continues to the end of document, or until the printer-pause code (40H) is detected. At a pause, printing can either be aborted (caps & symbol shifts together) or restarted (any other keypress). A printer pause can also be forced at any time by pressing any key.

2K HEX DUMP

8000	C3	F0	80	C3	29	81	C3	4B
8008	85	92	41	00	C5	F5	F5	01
8010	DF	FC	ED	78	CB	47	20	FA
8018	F1	01	DF	FE	ED	79	01	DF
8020	FC	3E	04	ED	79	C5	C1	3E
8028	06	ED	79	F1	C1	C9	00	00
8030	00	00	00	00	00	00	00	00
8038	00	00	00	00	00	00	01	00
8040	32	39	31	32	33	00	00	00
8048	00	00	00	00	00	27	21	00
8050	00	00	00	00	40	F0	F9	2C
8058	88	00	04	2C	88	2D	88	20
8060	C3	0C	80	FF	20	FF	10	08
8068	04	FE	04	08	10	00	00	00
8070	00	00	00	00	00	00	08	10
8078	20	7F	20	10	08	00	09	02
8080	83	08	10	83	0B	3B	83	0A
8088	6F	83	07	86	83	36	77	83
8090	37	4A	83	0D	81	83	38	94
8098	83	35	9C	83	49	32	84	69
80A0	E9	83	44	A4	83	64	AC	83
80A8	51	8B	83	41	43	83	63	E1
80B0	83	5A	24	84	42	05	82	50
80B8	C0	85	45	FF	84	47	F1	81
80C0	4D	20	85	56	EC	81	53	F6
80C8	81	43	4A	86	55	EE	86	0E
80D0	00	87	46	13	87	06	F7	86
80D8	4A	D7	81	53	D7	81	58	D7
80E0	81	57	DA	87	48	DF	87	4C
80E8	D7	81	52	D7	81	00	D7	81
80F0	00	3E	01	32	0A	5C	FD	CB
80F8	30	9E	11	00	FB	21	00	3D
8100	01	00	03	ED	B0	3E	CF	01
8108	DF	FD	ED	79	3E	F9	ED	79
8110	3E	0F	01	DF	FF	ED	79	3E
8118	06	01	DF	FC	ED	79	11	00
8120	88	21	38	80	01	2E	00	ED

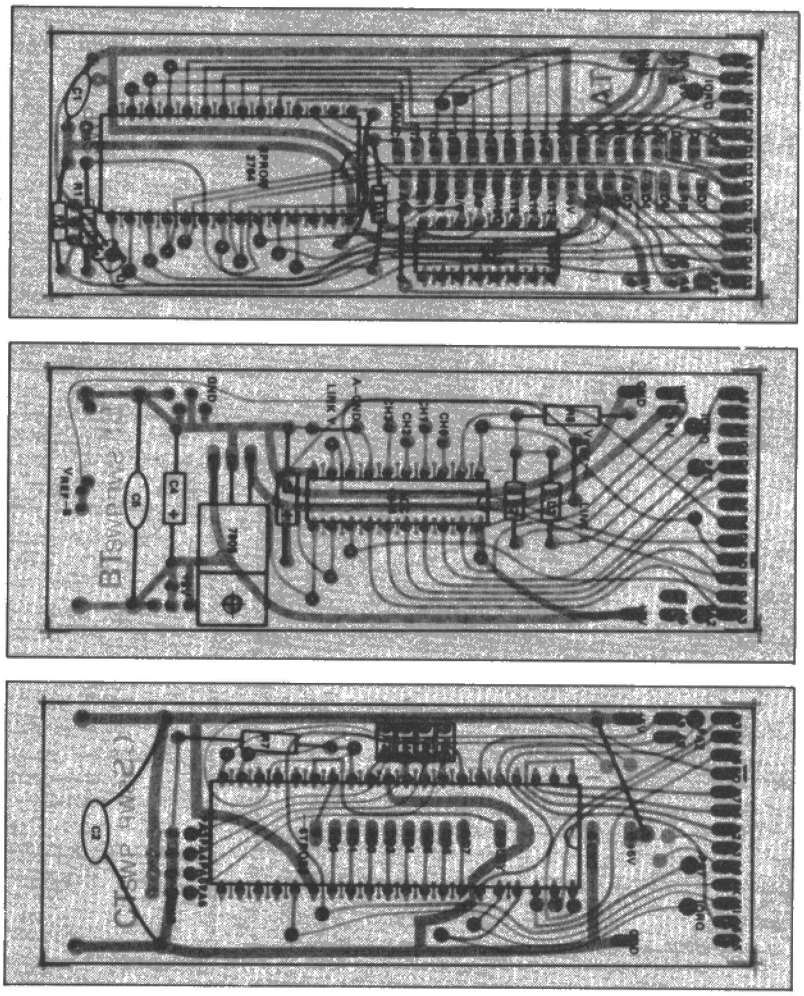
2K HEX DUMP (continued)

8128	B0	DD	21	1A	88	FD	CB	01	8370	23	88	CD	52	83	18	CF	2A	85B8	0E	C8	CD	C5	82	23	18	06
8130	AE	3E	02	CD	01	16	01	00	8378	1F	88	CD	52	83	22	1F	88	85C0	00	00	00	2A	23	88	11	9B
8138	00	CD	B6	82	04	CD	B6	82	8380	C9	CD	77	83	18	E9	CD	4A	85C8	85	CD	DE	86	7E	FE	FF	C8
8140	01	21	18	CD	D9	0D	11	97	8388	83	18	B0	21	2C	88	22	1F	85D0	DD	46	02	B8	28	D5	CD	28
8148	81	01	20	00	CD	E1	86	01	8390	88	C3	46	83	06	0A	CD	02	85D8	88	FE	0D	20	14	DD	CB	00
8150	21	14	CD	D9	0D	01	21	14	8398	83	10	FB	C9	06	0A	CD	10	85E0	56	28	0E	3E	0A	CD	28	88
8158	CD	D9	0D	11	B7	81	01	20	83A0	83	10	FB	C9	CD	AC	83	CA	85E8	DD	CB	00	5E	28	03	CD	28
8160	00	CD	E1	86	00	DD	CB	00	83A8	4A	83	18	F8	21	2C	88	ED	85F0	88	23	FD	CB	01	6E	20	B3
8168	66	CC	52	82	DD	CB	00	6E	83B0	5B	23	88	B7	ED	52	C8	CD	85F8	18	D2	2A	23	88	11	2C	88
8170	F5	CC	FA	85	F1	CC	38	86	83B8	D1	83	62	6B	1B	D5	ED	B0	8600	B7	ED	52	44	4D	03	19	2B
8178	CD	C5	82	CD	65	85	21	7E	83C0	2A	25	88	2B	22	25	88	E1	8608	3E	0D	11	00	00	ED	B9	20
8180	80	5F	7E	23	B7	28	07	BB	83C8	2B	3E	0D	BE	23	22	23	88	8610	06	E2	17	86	13	18	F6	EB
8188	28	04	23	23	18	F4	5E	23	83D0	C9	E5	ED	4B	23	88	2A	25	8618	23	22	3E	80	2A	3E	80	CD
8190	56	EB	11	65	81	D5	E9	42	83D8	88	B7	ED	42	4D	44	03	E1	8620	B0	86	01	0F	17	CD	D9	0D
8198	2D	46	52	45	45	20	20	20	83E0	C9	CD	AC	83	CD	65	85	18	8628	11	40	80	01	05	00	CD	E1
81A0	57	4F	52	44	53	20	20	20	83E8	03	CD	65	85	21	2C	88	ED	8630	86	01	20	15	CD	D9	0D	C9
81A8	20	43	55	52	2D	4C	20	20	83F0	5B	23	88	13	B7	ED	52	C8	8638	2A	1D	88	ED	5B	25	88	B7
81B0	20	20	54	4F	54	2D	4C	2D	83F8	F0	2A	1D	88	ED	5B	25	88	8640	ED	52	CD	B0	86	01	21	17
81B8	2D	2D	2D	2D	2D	2D	2D	2D	8400	B7	ED	52	C8	62	6B	13	CD	8648	18	DB	CD	A2	86	ED	5B	23
81C0	2D	2D	2D	2D	2D	2D	2D	2D	8408	D1	83	ED	B8	2A	23	88	77	8650	88	DD	36	01	00	21	00	00
81C8	2D	2D	2D	2D	2D	2D	2D	2D	8410	23	22	23	88	2A	25	88	23	8658	01	00	00	13	1A	FE	FF	28
81D0	2D	2D	2D	2D	2D	2D	2D	01	8418	22	25	88	C9	49	4E	53	45	8660	21	FE	40	28	1D	FE	0D	28
81D8	20	15	CD	D9	0D	11	E4	81	8420	52	54	20	20	2A	25	88	2B	8668	15	FE	20	38	EE	FE	22	38
81E0	CD	DE	86	C9	43	4F	4D	4D	8428	22	23	88	CD	21	83	22	1F	8670	0E	3E	01	DD	BE	01	DD	77
81E8	41	4E	44	3F	01	02	00	E1	8430	88	C9	11	1C	84	CD	DE	86	8678	01	28	E0	23	18	DD	03	AF
81F0	C9	01	01	00	E1	C9	2A	25	8438	DD	CB	00	6E	CC	FA	85	DD	8680	18	F1	C5	CB	3C	CB	1D	CD
81F8	88	23	23	11	1A	88	B7	ED	8440	CB	00	6E	F5	CC	1C	86	F1	8688	B0	86	01	18	17	CD	D9	0D
8200	52	44	4D	E1	C9	01	00	00	8448	CC	38	86	DD	CB	00	66	CC	8690	11	40	80	01	05	00	CD	E1
8208	E1	C9	C5	E5	D5	18	10	C5	8450	52	82	CD	65	85	FE	05	28	8698	86	E1	CD	B0	86	01	06	17
8210	E5	D5	C5	01	00	FA	26	00	8458	7F	FE	0D	20	07	2A	3E	80	86A0	18	83	26	00	DD	6E	0D	CD
8218	6F	29	29	29	09	EB	C1	CD	8460	23	22	3E	80	FE	5F	20	1F	86A8	B0	86	01	18	15	C3	25	86
8220	2E	82	06	08	1A	77	24	13	8468	2A	23	88	E5	23	CD	21	83	86B0	11	40	80	DD	E5	DD	21	D4
8228	10	FA	D1	E1	C1	C9	C5	78	8470	D1	B7	ED	52	7D	E6	07	20	86B8	86	3E	2F	DD	4E	00	DD	46
8230	E6	18	C6	40	67	78	E6	07	8478	02	3E	08	47	3E	20	C5	CD	86C0	01	C6	01	ED	42	30	FA	09
8238	0F	0F	0F	81	6F	C1	C9	00	8480	EC	83	C1	10	F7	18	3A	21	86C8	12	DD	23	DD	23	13	0D	20
8240	11	45	80	C5	CD	2E	82	06	8488	38	84	E5	FE	0C	CA	AC	83	86D0	E8	DD	E1	C9	10	27	E8	03
8248	08	7E	12	24	13	10	FA	C1	8490	FE	08	CA	3B	83	FE	09	CA	86D8	64	00	0A	00	01	00	01	08
8250	00	C9	00	2A	23	88	CB	FE	8498	02	83	FE	08	CA	10	83	FE	86E0	00	E5	78	B1	0B	28	05	1A
8258	E5	2A	1F	88	06	02	0E	00	84A0	0A	CA	6F	83	FE	0E	CA	00	86E8	13	D7	18	F6	E1	C9	3A	1A
8260	ED	43	21	88	06	04	0E	00	84A8	87	FE	06	CA	F7	86	E1	FE	86F0	88	EE	20	32	1A	88	C9	FD
8268	04	DD	CB	00	4E	28	01	04	84B0	07	C8	FE	04	20	02	3E	5B	86F8	7E	30	EE	08	FD	77	30	C9
8270	3E	15	B8	38	3C	3E	20	B9	84B8	FE	0F	20	02	3E	5D	CD	EC	8700	3A	1A	88	EE	02	32	1A	88
8278	20	14	7E	FE	0D	28	0A	FE	84C0	83	ED	5B	21	88	3E	13	DD	8708	3E	02	CD	01	16	06	13	CD
8280	8D	20	07	ED	43	21	88	CB	84C8	CB	00	4E	20	01	3C	BA	D2	8710	44	0E	C9	01	20	15	CD	D9
8288	BE	23	0E	00	18	DA	7E	FE	84D0	3F	84	CD	77	83	C3	3F	84	8718	0D	11	CA	87	CD	DE	86	DD
8290	FF	CB	BF	28	15	CB	7E	28	84D8	CD	65	85	16	00	FE	30	28	8720	E5	2A	23	88	E5	DD	E1	3A
8298	06	ED	43	21	88	CB	BE	23	84E0	02	16	10	CD	65	85	FE	61	8728	27	88	47	0E	00	DD	23	DD
82A0	FE	0D	28	06	CD	0F	82	0C	84E8	38	06	FE	67	30	02	D6	07	8730	7E	00	FE	FF	28	67	FE	5D
82A8	18	CB	CD	B6	82	0E	00	18	84F0	D6	30	E6	1F	B2	18	C7	45	8738	28	54	3E	20	DD	BE	00	28
82B0	B7	E1	CB	BE	00	C9	C5	11	84F8	52	41	53	45	20	3F	20	11	8740	2F	3E	0D	DD	BE	00	28	35
82B8	6E	80	CD	0A	82	0C	3E	20	8500	F7	84	CD	DE	86	CD	65	85	8748	10	E3	79	B7	28	54	DD	23
82C0	B9	20	F4	C1	C9	E5	D5	C5	8508	FE	79	C0	2A	23	88	23	22	8750	DD	2B	3E	FF	DD	BE	00	28
82C8	01	00	02	CD	B6	82	11	20	8510	25	88	36	FF	2B	36	20	C9	8758	47	3E	20	DD	BE	00	20	F0
82D0	00	01	20	03	0D	C5	CD	2E	8518	4D	45	52	47	45	20	3F	20	8760	DD	BE	FF	28	EB	DD	BE	01
82D8	82	06	08	7E	B7	ED	52	77	8520	11	18	85	CD	DE	86	CD	65	8768	28	E6	DD	36	00	0D	18	B7
82E0	19	24	10	F7	C1	B1	20	EC	8528	85	FE	79	C0	21	2C	88	22	8770	DD	BE	FF	28	D3	DD	BE	01
82E8	01	01	03	CD	B6	82	0D	11	8530	23	88	CD	D1	83	ED	43	3C	8778	28	CE	0C	18	CB	DD	BE	FF
82F0	66	80	CD	0A	82	CD	D7	81	8538	80	ED	5B	1D	88	2A	25	88	8780	28	A5	DD	BE	01	28	A0	DD
82F8	01	20	15	CD	D9	0D	C1	D1	8540	ED	B8	ED	53	3A	80	01	03	8788	36	00	20	0C	18	BA	DD	23
8300	E1	C9	2A	25	88	ED	5B	23	8548	00	E1	C9	ED	5B	25	88	1B	8790	DD	7E	00	FE	FF	28	06	FE
8308	88	13	B7	ED	52	C8	18	0C	8550	2A	3A	80	23	ED	4B	3C	80	8798	5B	20	F3	18	8A	DD	E1	C9
8310	21	2C	88	ED	5B	23	88	B7	8558	ED	B0	1B	ED	53	25	88	CD	87A0	DD	23	DD	E5	E1	22	23	88
8318	ED	52	C8	1B	ED	53	23	88	8560	8B	83	C3	29	81	E5	D5	C5	87A8	DD	E1	CD	21	83	22	1F	88
8320	C9	3E	0D	E5	01	2C	88	B7	8568	ED	4B	21	88	CD	3F	82	2A	87B0	CD	C5	82	01	20	15	CD	D9
8328	ED	42	28	0D	FA	39	83	44	8570	4D	80	23	22	4D	80	21	4E	87B8	0D	11	D2	87	CD	DE	86	CD
8330	4D	E1	2B	2B	ED	B9	23	23	8578	80	CB	56	28	05	11	76	80	87C0	52	82	CD	65	85	FE	20	20
8338	C9	E1	C9	2A	23	88	CD	21	8580	18	03	11	45	80	CD	0A	82	87C8	F9	C9	46	4F	52	4D	41	54
8340	83	18	03	2A	1F	88	22	23	8588	FD	CB	01	6E	28	E1	3A	08	87D0	20	20	4D	41	4E	55	41	4C
8348	88	C9	2A	1F	88	CD	21	83	8590	5C	FD	CB	01	AE	CB	BF	C1	87D8	4C	59	01	05	00	E1	C9	01
8350	18	2B	3E	0D	E5	ED	5B	25	8598	D1	E1	C9	50	52	49	4E	54	87E0	04	00	E1	C9	7F	52	2E	53
8358	88	1B	EB	B7	ED	52	28	09	85A0	49	4E	47	50	41	55	53	45	87E8	41	52	47	45	4E	54	20	31
8360	44	4D	E1	ED	B1	20	03	B7	85A8	44	20	20	CD	C5	82	11	A3	87F0	39	38	35	00	00	00	00	00
8368	C9	E1	CD	23	83	37	C9	2A	85B0	85	CD	DE	86	CD	65	85	FE	87F8	00	00	00	00	00	00	00	00

Finally, the sundries

It takes a few lines of BASIC to implement the SAVE, LOAD, MERGE and VERIFY functions. Note that the MERGE handled at line 18 is not the same as the Spectrum merge! To conserve machine code space, BASIC also services the LINE WIDTH setting. Control will pass back to the machine code sections automatically unless there is a tape-loading error or the BREAK key has been used (which is allowed). If you get stuck in BASIC in this way, pressing the RUN key will return you to the original cursor position. In **Listing 3** you will see no help messages, as even the most rudimentary of these take up about 4K of BASIC. However, they are very useful in some circumstances, and can be added as a MERGE to the core BASIC program. The cassette commands are easily changed to their Microdrive equivalents – the main wordprocessor code is assembled to take it well clear of the microdrive workspace and to leave room for the help messages. The machine code starts at 8000H, and runs to about 8800H, where the document area starts. The physical end of RAM for the document is F9FFH, though a POKE can set it lower if necessary. With this amount of RAM, the sign-on message for BYTES FREE should read around 29,000.

A cassette tape of the 2K RAM W.P., including the HELP messages and auxillary character shape table, is available for £4. *Next month,* the enhancements of the 8K EPROM version.



PCB overlays

SOFTWARE REVIEWS

Adam Denning reviews a selection of the latest BBC software.

ISO PASCAL

BBC Model B
Acornsoft
£69.00

Pascal is one of the most popular block structured languages in use at the moment, and the BBC Micro has been crying out for a good version of the language for an awfully long time. A few months ago, Acornsoft released its 'baby' S-Pascal, which is not much use to a serious programmer. Now that this has been followed up by the real Pascal, perhaps the BBC will get a much-needed new lease of life.

There are essentially three different standards for Pascal compilers, all of which are highly respected; they are 'Standard Pascal', as defined in the *User Manual and Report* by Jensen and Wirth, 'UCSD Pascal', as used in the University of California at San Diego's P-system operating system, and 'ISO Pascal', the most respected of the three. For a compiler to be ISO approved, it has to undergo an enormous number of tests. When I visited a company developing QL ISO Pascal the ISO test suite was brought out on show – it has more pages than all the year's newspapers put together! When, and only when, a compiler has passed all these tests can it be advertised as 'ISO approved'.

Acornsoft is one of the first companies to produce a small-machine approved Pascal. The Acornsoft system comes as two 16K ROMs, a disk and two manuals. The ROMs are for use in a standard BBC Micro which does not have a second 6502 processor attached; the disk has a version of the compiler suitable for the second processor. Pascal source is created using an in-built screen editor, and is then compiled into one of those 'compact interpretive codes' we hear so much about these days. This code can then be run with either the ROM or disk-based interpreter. Problem number one: it's unbelievably slow.

Compact codes, such as CINTCODE and P-code, have been popular for a while now as they allow 8-bit machines with 64K of RAM or less to run large programs. They also allow the compiler to be written for an 'idealised' machine, which makes the writing of the compiler itself rather easier. The code generation stage would be the same for each implementation and only the final interpreter needs to be altered. This tends to make the products easily portable, too. Compilation to pure machine code, although more efficient in the end, needs a larger code generator in the compiler, and the resultant code is also larger than its equivalent interpretive code system. With a meagre 32K of RAM on the Beeb, Acornsoft really had no choice. (There is talk of a secondary product which will compile the output from this compiler into 6502 machine code, but it has yet to see the light of day).

Another problem caused by the lack of space is that the ROM-based version is *not* ISO. It's a bit naughty of Acornsoft to claim that the product as a whole is ISO, as only the second processor version fulfils the standard. The difference between the two is trivial, it's true, but what's a standard for?

The manual supplied with the system is not as big as it looks, as most of it is a reproduction of the ISO specification with numerous appendices explaining the lack of conformance. What is there, though, is pretty essential stuff: it explains the memory useage of the compiler and interpreter and all the built-in commands.

The most interesting command is EDIT. The screen editor is easy to use and implements a rudimentary store file system; this allows both source and object code to be held in RAM at the same time, and the editor can edit RAM-based source. Although this is great if you haven't got disks, couldn't they have used the space taken up by all this to implement the missing bits of ISO? Acornsoft would no doubt say that this wasn't possible, as the interpreter and compiler occupy different ROMs.

Also included with the package is a tutorial guide to Pascal, called *Pascal From Basic*. The book is easy reading and uses characters like Professor Primple and Bill Mudd to carry across the idea of structured programming to great effect. I doubt if it would actually provide much of a 'Programming in Pascal' style text in its own right, though, as there isn't much of a reference section. An appendix in the Acornsoft manual is devoted to describing the differences between the compiler used in the book and the compiler supplied with the product! A lovely bit of co-ordination.

Programs developed within the Pascal environment may include optional tracing facilities, allowing a program to report the current equivalent source line number as it proceeds. Undoubtedly a useful function, as are most of the other compiler options. One option allows the non-ISO Acornsoft extensions to be used, things which allow operating system and direct memory access. Thus, it is possible to develop systems programming under ISO Pascal, but I find it an unlikely and unpleasant prospect, having used the tedious Acornsoft/TDI P-system implementation which is of course written in Pascal.

This version of Pascal, due no doubt to the inadequacies of the 6502 and the interpretive system used, is embarrassingly slow. Even running on the 6502 second processor, where one would expect it to perform at its best, it runs at around a quarter of the speed of Spectrum Pascal and between a third and a half the speed of (interpretive coded!) Computer One QL Pascal. Now that's bad.

A package for purists, then, rather than programmers. That'll set the cat among the pigeons!

BASIC COMPILER

BBC Model B
Pineapple Software
£25.00

For some reason, a truly good Basic compiler has never been produced for the BBC Micro. This may be because the interpreter is fairly fast, but it's more likely to be the fact that the 6502 processor is not easy to write compilers around.

Whenever companies do produce compilers, they all tend to suffer from the same problems: integer-only calculations, memory restrictions and only half the keywords implemented. This new compiler from Pineapple Software is a little better than most, but still begs the question 'Why now?' The BBC Micro had such wonderful potential, and if it had only got the software it needed on time Acorn wouldn't be in the mess it's in now. Acornsoft has the cheek to release its promised ISO Pascal compiler *three years* after the birth of the computer, and that turns out to be abysmally slow. The end is nigh, and it's far too late for Pineapple's compiler to make that much of a splash.

The compiler's main limitations are that only 26 integer variables may be used, and that string handling is fairly rough. The lack of variables can be alleviated by using local variables inside procedures and functions, but string manipulation has to be done in assembler. Pineapple intends to release a set of library routines to help programmers (but it should have thought of this before it released the compiler).

File handling is minimal, with recourse to assembler again being the only solution. Writing a recursive procedure to function using DEFPROC or DEFFN is perfectly legal, but as the compiled code uses the 256-byte 6502 hardware stack, there's not a lot of recursion you can get up to.

But this is silly. You can examine a Basic compiler and point out its limitations until the cows come home. The question is, does it have a market and does it fulfil its specifications? At the price Pineapple is selling it at, the program isn't overpriced, but I (this is a personal opinion) do not believe in buying a compiler at any price which only does half the job.

DIAGRAM

BBC Model B
Pineapple Software
£25.00

This program from Pineapple Software allows the BBC Micro owner to draw complicated diagrams and illustrations.

These pictures are not limited to one screen, and by using a blank disk to store each on it is possible to have diagrams which occupy as many as 39 screens. Two examples are given which demonstrate the technique rather nicely. The first is a plan of a house and the second is a circuit diagram. Although only one screen can be loaded, the cursor keys allow you to move anywhere within the diagram with ease.

Creation of diagrams is made reasonably simple by the inclusion of an online character designer. Once inside this, you can design and save new patterns for a selected number of characters, and subsequently use these characters in your diagrams. Text can be entered anywhere in a diagram, and each diagram is referenced by name.

Unfortunately, the program cannot be used very effectively as a 'painter' style program as there is no easy way to enter anything but lines, text and defined characters. Pineapple says that this is the price to pay for being able to store large diagrams on one disk. User defined characters are not limited to the usual 8 x 8 pixel grid, but can occupy 32 x 24 pixels if required. Everything is done in Mode 0 and a variety of printing routines are provided to allow you to dump out designs to a dot matrix printer.

Any part of a diagram may be 'indexed', allowing immediate access to that part of the diagram at a later date. The size of a diagram, in terms of screens occupied, may be altered at any time without the loss of previously entered information, and each can of course be saved to disk.

The whole program is fairly friendly, with menus throughout but as the thing only offers a limited facility and costs £25, it's a bit doubtful as to whether it's really that good a buy.

ATARI 130XE

Ken Alexander previews an 8-bit computer with plenty of software and twice the memory for your money.

'Power without the price', is the new slogan adopted by Atari to promote its new 130XE computer.

This machine is one of the first to be released by the new look Atari, now under the wing of former Commodore boss Jack Tramiel, and is the precursor of the eagerly awaited 68000 based ST models. While the 130XE is 'only an 8-bit micro' it offers features that set it aside from other machines in this class.

The 130XE represents a new generation of 128K (130,712 bytes – hence 130XE) 8-bit computers; the Commodore C128, expected 'soon' is another. Compare this with the new BBC model B Plus that shouts about its 64K RAM. The second attraction of the 130XE is its price, just £169.90. This is £20 below the current Commodore price for the 64 (although this is likely to come down in the near future) and less than half the price of the aforementioned B Plus.

The 130XE is not a 'new' design, more of an uprated version of the familiar Atari 800XL and the 400 before it. These machines were well respected for their sound and graphics capabilities but never achieved a significant penetration of the UK market, largely because the hardware was over-priced compared with the competition in the sector dominated by the Spectrum and the Commodore 64. The short history of the microcomputer market has shown that any new model entering the market at an uncompetively high price is almost certain to fail. Price cuts at a later stage have, to date, without exception proved ineffective. Pricing the 130XE at the aggressive level chosen by Atari means that the machine is in with a chance.

Another bonus for the computer is that there is plenty of compatible 800/400 software around, much of it games but some business titles are available as well. Anyone who requires a word processor, spreadsheet or data base will find a (limited) choice of suitable packages in the shops.

Hard wearing hardware

The 130XE has a very businesslike look and feel to it. The styling is very much in line with that of the ST computers. The case is finished in a neutral grey colour with cream keys sporting a tasteful brown ledgend – all very 'Habitat'. The 'feel' of the keyboard is good, excellent on a machine of this price. One minor niggle, a return key that is too small, but then this is a defect of so many low cost micros. Above the keyboard to the right hand side of the computer are five decently sized function keys labelled, from left to right, Help, Start, Select, Option and Reset. In keeping with the 'designer' look of the machine these keys are slanted at an angle that complements the ribbing on the top of the casing. Thus although the keys themselves are of adequate size it is easy to accidentally press two keys at once. This makes the Reset key rather vulnerable in its present position.

The I/O ports are situated along the rear edge of the 130XE, with the exception of two 9-pin D type joystick connectors on the right-hand side of the case.

Looking at the rear of the computer, the left most connector is given the all encompassing name of 'peripheral connector'. This 13-pin plug is in fact a serial I/O port and indeed most peripherals for the computer would be plugged in here. This includes printers (the machine has no dedicated Centronics printer port), cassette datacorder, modem and even the complementary disk drives. A very hardworking port indeed.

Next comes the cartridge port. This is a 30-pin connector that links most of the machine's address lines and the data bus to the outside world, along with the Read/

Write line, other assorted control signals and the +5V power line.

The enhanced cartridge interface (ECI) brings out the rest of the address lines (A13-A15) and some further control signals. Between them the cartridge slot and the ECI should allow virtually any device to be connected to the 130XE. The next two sockets provide outputs for a monitor and TV set respectively. The monitor socket provides only a composite video signal, no RGB signal is available.

The final socket is a DIN plug carrying the 5V supply from the external PSU. Next to this is the computer's on/off switch – a welcome feature for so many low cost computers fail to provide a way to elegantly 'power down'; it's left to the crude expedient of pulling the plug out.

The major components inside the 130XE are the familiar 6502 processor, the Antic VDG (an IC familiar to Atari users of old) the Pokey chip that handles the keyboard and provides the four-voice, 3.5 octave sound capability, and the Freddy custom memory management chip required because unaided the MPU can cope with only 64K of memory.

The firmware is contained in two ROMs: the OS has 16K of code but the (dated) Atari BASIC is given only 8K of space. Even allowing for the fact that some functions normally included within the BASIC ROM are in the OS ROM – the floating point routines for example – the small size of the BASIC interpreter explains why Atari BASIC is not the best around.

Banking on software

Atari BASIC is best described as workmanlike. That's not to say that it is seriously deficient in any way, it supports on-screen editing driven by the cursor control keys and has such niceties as checking of syntax as lines of a program are entered. The major drawback is that getting the machine to perform what many of today's BASIC programmers would consider mundane tasks requires recourse to machine code pokes. Not the sort of thing the tyro programmer would feel at home with.

The essential question is what use does the 130XE make of its extra 64K of memory? The answer, as far as those using the BASIC interpreter is concerned, is none. It is only the machine code programmer who will be able to utilise the full potential of all that storage space. The extra memory can also be used to store extra graphic screens, and version 2.5 of Atari DOS supports the use of the additional memory as RAM disk.

While on the subject of disks, Atari has adopted the 3.5" format. The ST computer will use the Sony drives and it seems that with 'home' as well as business machines going the Sony way, the battle of the minifloppy has been won – Amstrad are out in the cold with the 3" drive of the CPC664.

Should you buy?

There can be little doubt that the hardware to make or break Atari is the ST series. The XE is hardly likely to set the world alight. The market is saturated with 8-bit micros and it is difficult to see any new launch making a fortune for anyone. The Amstrad CPC464 and 664 computers are rapidly becoming the yardstick against which 8-bit machines are measured and, although the XE is good value for money, it still doesn't beat the CPC464 in performance per pound. The extra memory looks good on the spec sheet but in practice will be of little benefit to most users. The good software base for the XE, by virtue of its 400/800 parenthood, is a good selling point, but then the Amstrad software industry is rapidly growing. The XE must be seen primarily as a games machine that enters an over crowded market. Chances are it will fail to make any impact in the UK. A shame because it is a nice machine.

DATAFILE: ATARI 130XE

PROCESSOR:	6502
MEMORY:	131,072 (128K)
VIDEO:	Choice of 256 colours in 11 graphics modes.
SOUND:	Four independent voices
I/O:	Two joystick ports, cartridge slot, serial bus, expansion port, RF and composite video outputs
PRICE:	£169.99